

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Recherche d'instruments de mesure des performances du système MUMPS - 11

Bardez, Luc

*Award date:*  
1975

*Awarding institution:*  
Université de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX A NAMUR.

INSTITUT D'INFORMATIQUE.

ANNEE ACADEMIQUE 1974-1975.

Institut d'Informatique  
Bibliothèque  
Tél. 081-747.49 FNDP NAMUR

RECHERCHE D'INSTRUMENTS DE MESURE  
DES PERFORMANCES DU SYSTEME MUMPS-11.

LUC BARDEZ.

MEMOIRE PRESENTE EN VUE DE L'OBTENTION  
DU GRADE DE MAITRE EN INFORMATIQUE.

Que Monsieur F. BODART, Directeur de l'Institut d'Informatique, ainsi que les membres de ce même Institut, trouvent ici l'expression de mes remerciements pour la formation acquise au cours de ces deux années de Maîtrise en Informatique.

J'exprime également mes remerciements à Monsieur S. de HEPCEE qui a bien voulu assurer la direction de ce mémoire.

Que Monsieur R.J. ROTHSTEIN trouve ici l'expression de ma reconnaissance pour avoir permis mon séjour à la société Interactive Systems.

Que Monsieur M.E. LEVIN soit assuré de ma gratitude pour l'aide patiente et les conseils prodigués durant la réalisation de ce travail.

Namur, le 30 juillet 1975.



## T A B L E D E S M A T I E R E S

- TABLE DES MATIERES
- INTRODUCTION
- CHAPITRE I : LE PROBLEME DE LA MESURE DES PERFORMANCES
  - 1.1. Concept de performance
  - 1.2. Evaluation de la performance
- CHAPITRE II : EVALUATION DE LA PERFORMANCE DE L'EXECUTIVE MUMPS-11
  - 2.1. Aspects fondamentaux de l'EXECUTIVE MUMPS-11
  - 2.2. Description des objectifs et des conditions initiales du travail
  - 2.3. Description du programme de surveillance.
  - 2.4. Présentation d'outils complémentaires d'analyse
  - 2.5. Conclusions
- CHAPITRE III : LIMITES ET RECHERCHE D'AMELIORATION DU PROGRAMME DE SURVEILLANCE
  - 3.1. Mesure du temps d'exécution de la procédure extractive
  - 3.2. Mesure du temps d'exécution du programme de surveillance
  - 3.3. Recherche d'améliorations
  - 3.4. Surveillance d'une même charge par deux versions du programme
  - 3.5. Etude des variations du temps d'exécution
  - 3.6. Résultats parallèles
- CHAPITRE IV : EVALUATION DU SERVICE RENDU PAR LE SYSTEME MUMPS-11
  - 4.1. Introduction
  - 4.2. Mesure du temps de réponse
  - 4.3. Rôle de la méthode d'accès à la "data base" MUMPS-11
- CONCLUSION
- APPENDICE A : PRESENTATION DES INFORMATIONS ET RESULTATS OBTENUS A LA BANQUE DE BRUXELLES
- APPENDICE B : "MUMPS" : BUTS ET HISTORIQUE
- APPENDICE C : "MUMPS -11 SYSTEM"
- APPENDICE D : DESCRIPTION DU LANGAGE MUMPS-11



- APPENDICE E : IMPERFECTIONS DE MUMPS - 11

- BIBLIOGRAPHIE

## I N T R O D U C T I O N

=====

Depuis le début des temps, l'homme a constamment exercé sa sagacité et son ingéniosité dans deux directions différentes, souvent complémentaires.

Dans la première direction, l'homme aspirait à la compréhension des phénomènes qu'il découvrait, dans la seconde, il recherchait des techniques nouvelles destinées à alléger son travail. Quel qu'ait pu être la direction choisie, il a été régulièrement confronté avec les quatre questions suivantes :

- a) quel est le problème?
- b) comment ce problème peut-il être résolu?
- c) quel est le champ d'application d'une solution?
- d) quel est l'efficacité d'une solution donnée?

Si l'application de la Méthode Scientifique a efficacement aidé l'homme à répondre à ces questions, il faut cependant constater qu'en ce qui concerne la technique informatique, le niveau général d'instrumentation et de mesure est encore dans un état primitif (selon l'auteur de la référence [1]).

En consultant les articles écrits à ce sujet, on peut observer qu'en général, la mesure de performances est surtout stimulée par des motifs économiques tels que: réduction d'une configuration, ajournement d'une commande, choix d'un nouveau matériel, évaluation comparée de plusieurs offres, ...

En fait, quel que soit le motif cité pour procéder à une mesure de performances d'un système, une évaluation est toujours utile pour éclairer la complexité des modules d'un système et leurs relations. De plus, et surtout depuis l'existence des systèmes d'exploitation, il s'agit d'avoir la possibilité de dominer un produit élaboré de manière à ce que, quel que soit la circonstance, on puisse toujours prendre une décision consciente et fixer les choix utiles pour, par exemple, permettre à un système de fonctionner dans des conditions pour lesquelles il a été conçu [2].

On se placera, pour ce mémoire, dans le cas concret du système MUMPS-11. La première partie du travail traitera de l'évaluation de la performance de son Executive, quant à la seconde, elle traitera plus spécialement de l'évaluation du service rendu par le système MUMPS-11.

## CHAPITRE I :

---

### LE PROBLEME DE LA MESURE DES .....

#### PERFORMANCES. .....

1.1. Concept de performance.

1.2. Evaluation de la performance.



### 1.1. CONCEPT DE PERFORMANCE.

La performance d'un système peut se définir comme étant l'efficacité constatée de celui-ci, dans l'exécution d'une tâche pour laquelle il a été conçu [2]. Cette approche fait apparaître clairement le caractère relatif de la mesure de performances.

La performance peut à la fois être qualifiée et quantifiée. Il faut donc disposer, pour cela, d'un ensemble de paramètres ou variables dont l'ensemble des valeurs traduira l'efficacité du système. Il y a deux façons d'envisager la mesure des performances. La première manière consiste à considérer le système comme une "boîte noire" dont on ignore les mécanismes. La mesure consiste alors à charger le système et à analyser les conditions d'obtention des résultats. L'évaluation du débit ("throughput") est classique, selon cette optique. La seconde manière de procéder à la mesure de performances s'oppose à la précédente, puisqu'elle s'applique à démontrer les mécanismes du système. La charge constitue, selon cette optique, une séquence d'appels aux ressources du système. Des variables telles que la durée d'occupation d'une ressource, le nombre d'appels à une ressource, sont classiques dans cette façon d'envisager la performance.

Si ces deux méthodes apparaissent très différents, elles n'en sont pas moins naturelles si d'une part, c'est un utilisateur qui envisage l'évaluation de performances, et si d'autre part, c'est un concepteur ou un gestionnaire de système qui l'examine.

L'évaluation des performances est soumise à des limites économiques qui, si elles ne sont pas théoriquement impossibles à éliminer, ne peuvent être surmontées qu'à un prix dont le rapport coût/efficacité serait très mauvais. C'est notamment le cas de la recherche d'algorithmes optimaux qui, s'ils réalisent l'optimum dans le champ pour lequel ils ont été conçus, ne se justifient pas vis-à-vis des moyens financiers dont on dispose. Les frais de recherche, d'implémentation et d'exécution de tels algorithmes ne peuvent, généralement, pas être couverts par le budget de fonctionnement d'une installation.



## 1.2. EVALUATION DE LA PERFORMANCE

Les techniques de mesure des performances d'un ordinateur ont principalement pour objet l'amélioration des performances et la réduction des coûts. [3] .

Le problème de l'évaluation se pose de manières différentes selon qu'il s'agit du constructeur ou de l'utilisateur. Selon qu'il s'agit d'un constructeur, la mesure des performances sera surtout envisagée dans le but de pouvoir prédire les conséquences éventuelles de décisions prises au niveau du "design".

Selon qu'il s'agit maintenant de l'utilisateur, l'évaluation de performances contribuera efficacement à la réalisation d'objectifs tels que :

- . La réduction de la configuration.
- . Le réglage ("tuning") d'un système existant.
- . Le choix d'un nouveau matériel.
- . L'évaluation composée de plusieurs offres.
- . L'ajournement d'une commande.

Les principales techniques d'évaluation sont les suivantes

[4], [5] :

- a) Les modèles de simulation.  
C'est certainement la méthode la plus souple et la plus puissante. Deux types de simulation peuvent être utilisés pour la mesure de performance d'un système. Il y a d'une part les modèles de simulation orientés événements. Ces modèles fournissent des distributions de probabilité qui décrivent la performance de composantes du système à évaluer. D'autre part, il y a les modèles de simulation utilisant des données déduites d'investigation empiriques préalables.
- b) Le monitoring (hardware et software).  
Le monitoring est une méthode basée sur la collection de données caractéristiques des performances d'un système existant. Cette méthode est généralement utilisée pour localiser les "bottlenecks" qui limitent la performance d'un système.
- c) Les modèles analytiques.  
Ces modèles sont constitués par une représentation mathématique du système.
- d) Les "benchmarks" : ce sont des programmes existants, rédigés dans un langage donné et qui sont exécutés sur l'installation dont on veut avoir un ordre de grandeur de la performance .
- e) Les programmes synthétiques .  
Ces programmes sont analogues aux " benchmarks" et utilisés dans le même but. La seule différence réside dans le fait qu'il ne sont pas nécessairement existants avant la décision de procéder à une évaluation de performances.

## C H A P I T R E II :

### EVALUATION DE LA PERFORMANCE

.....

#### DE L'EXECUTIVE MUMPS-11

.....

- 2.1. Aspects fondamentaux de l'Exécutive MUMPS-11.
- 2.2. Description des objectifs et des conditions initiales du travail.
- 2.3. Description du programme de surveillance.
- 2.4. Présentation d'outils complémentaires d'analyse.
- 2.5. Conclusions.

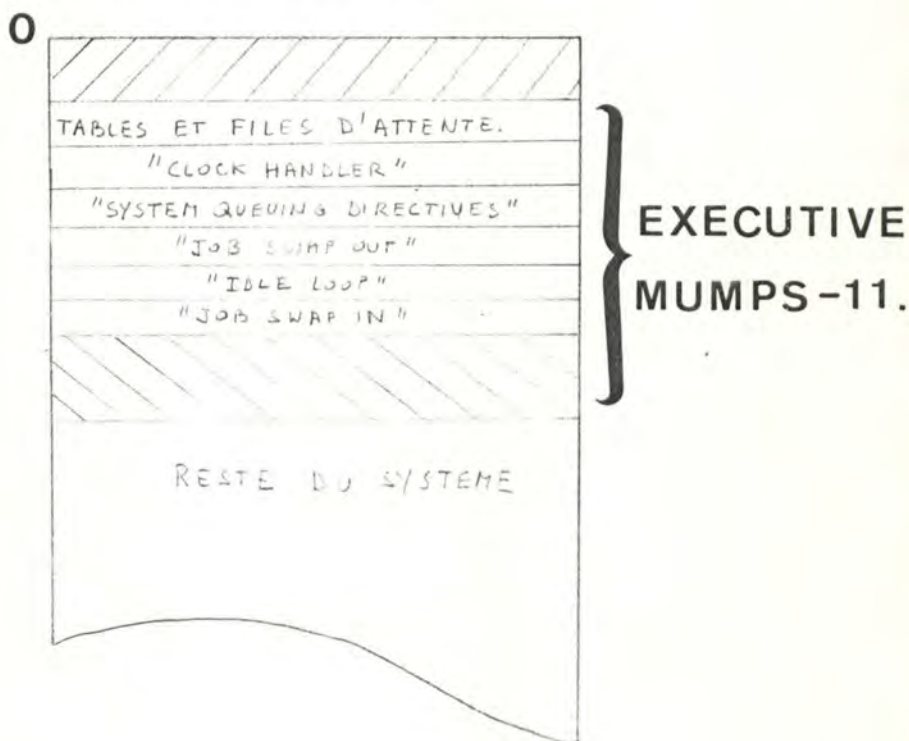


## 2.1. ASPECTS FONDAMENTAUX DE L'EXECUTIVE MUMPS - 11.

MUMPS - 11 [6], [7], [8], [9] désigne à la fois un langage interprétatif de haut niveau et un système d'exploitation orienté time-sharing. Les aspects du time-sharing sont implémentés dans le module "EXECUTIVE" de système d'exploitation. Le module "EXECUTIVE" permet la multiprogrammation par partitions, au moyen de l'assignation dynamique d'espace mémoire de taille fixe aux utilisateurs.

L'Exécutif MUMPS - 11 est fondamentalement subordonné aux E/S. En effet, le moment où un job sera exécuté et le temps qui lui sera accordé, sont essentiellement basés sur l'activité I/O de ce job. Ces principes sont implémentés par l'intermédiaire d'un système hiérarchisé de files d'attente.

### 2.1.1. Configuration de la mémoire.



On trouvera une description détaillée des tables et des "system queuing directives" en appendice C

### 2.1.2. Description des files d'attente du système.

2.1.2.1. Définition : les différentes files d'attente du système sont les suivantes :

- a) "RUN QUEUE" (RUNQ) : contient le numéro du job qui a le contrôle du processeur. Cette file d'attente est toujours réduite à un élément.

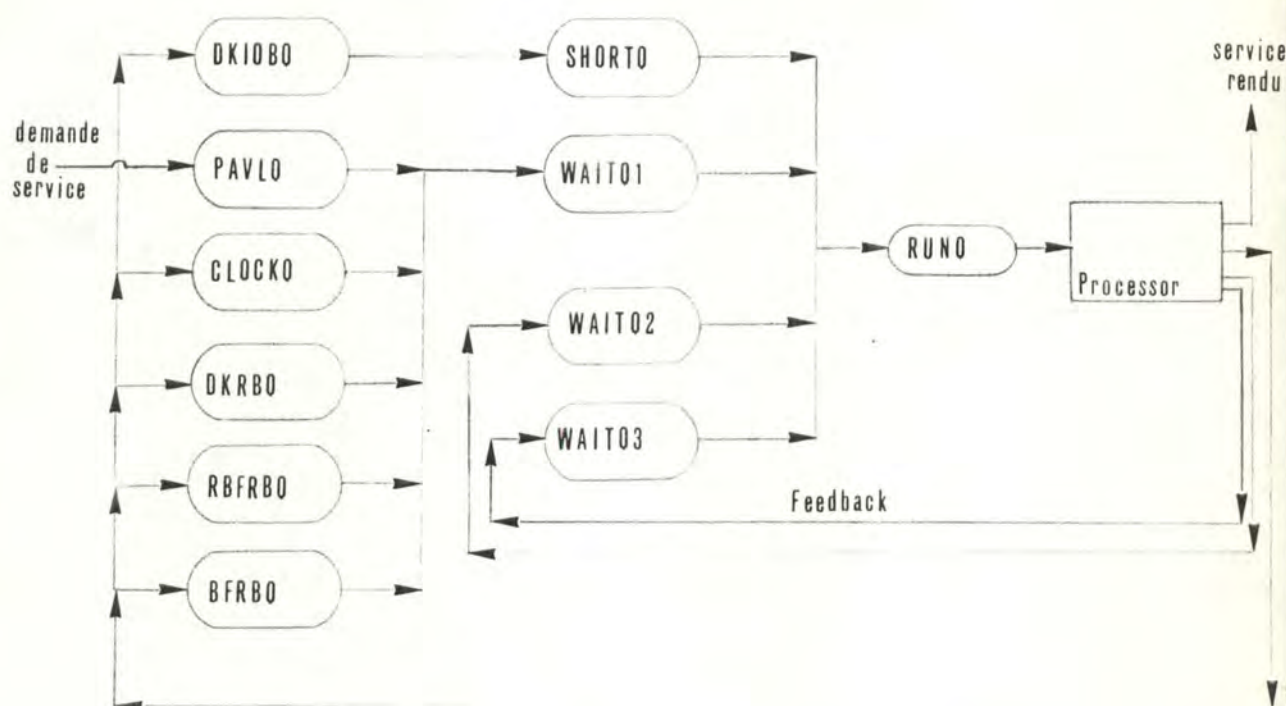
- b) "SHORT QUEUE" (SHORTQ) : cette file d'attente est destinée à contenir le job qui vient juste de terminer une E/S disque. C'est la "WAIT QUEUE" de plus haute priorité. La tranche de temps allouée à un job issu de la SHORT QUEUE est de 1 tick (1 seconde = 50 ticks    1 tick = 20 millisecondes).
- c) "DISK I/O BOUND QUEUE" (DKIOBQ) : file d'attente qui contient le numéro du job qui réalise actuellement un accès disque logique.
- d) "DISK RESOURCE BOUND QUEUE" (DKRQB) : file d'attente des jobs qui ont besoin de la ressource disque pour continuer leur exécution.
- e) "RING BUFFER RESOURCE BOUND QUEUE" : (RBFQB) liste des jobs actuellement en attente d'un "ring buffer" de 64 caractères.
- f) "256 - WORD BUFFER RESOURCE BOUND QUEUE" (BFRQB) : liste des jobs actuellement en attente d'un buffer de 256 mots en vue de réaliser une E/S autre que disque.
- g) "HIGH PRIORITY WAIT QUEUE" (WAITQ1) : liste des jobs en attente du processeur. Ces jobs recevront une tranche de temps (5 ticks) lorsqu'ils recevront le contrôle du processeur. WAITQ1 suit immédiatement SHORTQ dans l'ordre des priorités.
- h) "MEDIUM PRIORITY WAIT QUEUE" (WAITQ2) : liste des jobs en attente du processeur. Ces jobs recevront deux tranches de temps (10 ticks).
- i) "LOW PRIORITY WAIT QUEUE" (WAITQ3) : liste des jobs en attente du processeur. Ces jobs recevront trois tranches de temps (15 ticks).
- j) "CLOCK QUEUE" (CLOCKQ) : file d'attente des jobs suspendus pour une période de temps déterminée à la suite des commandes HANG ou "timed READ".
- k) "PARTITION AVAILABLE QUEUE" (PAVLQ) : file d'attente des partitions disponibles.

La manipulation de ces files d'attente est réalisée par un ensemble de routines constituant le module "System Queuing Directives" décrit de manière détaillée dans la seconde partie de l'appendice C.

#### 2.1.2.2. Ordonnancement des files d'attente.

Il peut être schématisé de la façon suivante:





Pour compléter ce schéma, on signalera que les jobs qui réalisent une E/S autre que disque, sont placés en "HUNG STATE" (voir description de JOBTAB en appendice c), que ces jobs sont libérés par interruption et placés en fin de WAITQ1.

Cette politique d'ordonnancement favorise l'aspect interactif d'un job en suivant très rapidement les tâches subordonnées à une E/S. On peut dire que les jobs ayant une forte "I/O activity" passent souvent et rapidement en "RUN QUEUE".

Les jobs ayant une forte "CPU activity" passent peu mais plus longtemps.

### 2.1.2. 3. Module "IDLE LOOP"

Les diverses priorités que reçoit un job au cours de son traitement se déduisent des diverses files d'attente par lesquelles il transite.

Les différentes files d'attente décrites en 2.1.2.1. l'ont été dans l'ordre des priorités décroissantes, à savoir :

- . "SHORT QUEUE" (SHORTQ)
- . "DISK I/O BOUND QUEUE" (DKIOBQ)
- . "DISK RESOURCE BOUND QUEUE" (DKRBQ)
- . "RING BUFFER RESOURCE BOUND QUEUE" (RBFRBQ)
- . "256-W BUFFER RESOURCE BOUND QUEUE" (BFRBQ)
- . "HIGHEST PRIORITY WAIT QUEUE" (WAITQ1)
- . "MEDIUM PRIORITY WAIT QUEUE" (WAITQ2)
- . "LOW PRIORITY WAIT QUEUE" (WAITQ3)



L'algorithme d'ordonnancement (IDLE LOOP) consiste à analyser des files d'attente pour déterminer le job qui sera exécuté. En plus, si aucun job n'est trouvé, il sera éventuellement fait appel au "garbage collector" pour collecter un bloc disque s'il en existe un disponible et si la "disk I/O bound queue" est vide.

Si on ne trouve aucun job dans les files d'attente et aucun bloc disque à désallouer, "IDLE LOOP" Répète indéfiniment son processus.

### 2.1.3. Les autres modules de l'exécutive.

2.1.3.1: Les tables contiennent les informations de base du système, et tout ce qui concerne et permet le contrôle d'un job dans le système. Une parmi les plus importantes est la "JOBTABLE" qui sert également d' "entry space" pour les files d'attente du système. Les tables sont présentées de manière détaillée dans la première partie de l'appendice C.

2.1.3.2. Le manipulateur d'horloge ("Clock Handler").  
La fonction du "Clock handler" consiste à traiter les interruptions générées par l'unité KW11-L ("LINE FREQUENCY CLOCK"). L'intervalle séparant deux interruptions successives de "KW11-L" est de 1 tick soit 20 millisecondes. L'unité de temps du système MUMPS-11 est donc de 0.02 sec'.  
A chaque interruption, différentes entrées de SYSTAB (voir appendice C) sont mises à jour pour tenir compte du temps qui s'écoule.  
En plus de cela, tous les 50 ticks (= 1sec), le manipulateur d'horloge analyse la file d'attente "Clock queue".  
Au cours de cette analyse, on décrémente d'une unité le temps de suspension des jobs en "Clock queue" et on sort de cette file d'attente le job pour laquelle temps de suspension est devenu nul.

2.1.3.3. Les modules "SWAP OUT" et "SWAP IN".  
Le module "JOB SWAP OUT" a pour but de sortir le job qui est dans la "RUN QUEUE" et à sauver toutes les informations nécessaires à un redémarrage ultérieur.  
Le module "JOB SWAP IN" réalise logiquement le contraire du module "JOB SWAP OUT".

le de reboisement n'est pas une marchandise :  
on ne peut ni le vendre, ni l'acheter.  
Il n'est pas non plus un objet faisant l'objet d'un  
sur une même surface déserte :  
il n'est pas fait pour être donné en spectacle.  
Il n'est pas enfin un objet qu'on transporte  
comme on le place un verre sur une table.  
Il a des racines de certains arbres,  
il ne peut être transplanté.



## 2.2. Description des objectifs et des conditions initiales du travail.

### 2.2.1. Description des objectifs.

D'une manière générale, lorsqu'on aborde le problème de la mesure de performances, on peut constater que, si de gros efforts sont réalisés pour dégager qualitativement les mérites d'architectures nouvelles [1-], en revanche, peu d'efforts sont consacrés à l'analyse quantitative des performances du système.

Ce manque de mesures quantitatives est essentiellement dû au fait que l'analyse d'un système peut seulement être réalisée à partir d'investigations qui tiennent compte d'un ensemble plus ou moins compliqué d'interactions entre le hardware et software. (Le système des interruptions est un exemple d'ensemble d'interactions entre hardware et le software.) La performance est fonction non seulement des caractéristiques des programmes d'applications (besoins en mémoire, I/O, traitement) mais aussi dépend de caractéristiques hardware/software, telles que "l'Overhead" du système d'exploitation, la politique d'ordonnancement des files d'attente, la vitesse de la mémoire secondaire, la vitesse de traitement d'une instruction, etc...

En général, on distingue trois grandes catégories de méthodes d'analyse des systèmes informatiques.

Ce sont le modèle mathématique [11], la simulation [12], et l'investigation empirique [13],

C'est cette troisième méthode d'analyse qui sera adoptée pour la recherche de l'évaluation des performances du système MUMPS-11.

L'investigation empirique peut être envisagée sous trois points de vue :

- a) l'instrumentation externe ("hardware monitoring"): ce procédé consiste à connecter des sondes à divers points de la machine. Les sondes sont conçues pour mesurer l'activité électrique (en nombre d'impulsions enregistrées) en ces points. La mesure du pourcentage de temps "actif" d'un canal, est l'une des mesures classiques du monitoring hardware [14].
- b) L'instrumentation interne ("software monitoring"): un moniteur software est un ensemble de programmes destinés à examiner, par coups de sondes, le contenu des divers registres et tables du système. [15, 16]
- c) La combinaison d'instrumentations interne et externe : c'est probablement le meilleur type d'instrumentation qui rassemble la souplesse du "software monitoring" et le caractère indépendant du "hardware monitoring". [17]



En ce qui concerne le présent travail, le point de vue utilisé dans le cadre de l'investigation empirique, sera celui de l'instrumentation interne.

La méthode d'analyse ainsi précisée, il reste à fixer le champ auquel on va l'appliquer. L'Exécutive MUMPS-11 a été choisi à cet effet. Dans cet Exécutive, c'est plus particulièrement la stratégie de l'ordonnement des files d'attente qui sera étudiée. A cet effet, on va construire un échantillon d'états du système. Un état du système sera constitué de l'ensemble des états de chaque partition. On espère ainsi obtenir une mesure du taux d'utilisation de chaque ressource de l'installation et éventuellement de détecter la ou les ressources dont l'utilisation donne naissance à des goulots d'étranglement ("bottlenecks") dans le système.

Cet aspect (observation du système) peut être complété par l'aspect "réglage" de l'Exécutive. Le réglage d'un système peut être réalisé par l'intermédiaire de paramètres de contrôle. Le paramètre le plus important de l'Exécutive est la valeur (en nombre de ticks) que l'on attribue à une tranche de temps standard.

Une modification de la valeur de ce paramètre peut engendrer en ce qui concerne la performance, des différences qui pourront apparaître dans les résultats de la surveillance de l'état du système. Ce travail est réalisable : le langage MUMPS-11 fournit tous les éléments nécessaires à cette réalisation.

D'autre part, lorsqu'on est confronté au problème de la mesure de performances d'un système d'exploitation orienté time-sharing, il est également intéressant d'avoir une mesure du temps d'"Overhead" du système. Dans le cas de MUMPS-11, l'intérêt porterait plus particulièrement sur la mesure de l'"Overhead" de l'Executive. On peut envisager cette mesure en deux étapes :

- a) obtenir une valeur moyenne du temps d'exécution des divers modules et procédures de l'Executive.
- b) Compter pour chaque job le taux d'utilisation des divers modules et procédures.

Ce n'est toutefois, pas réalisable dans le langage MUMPS-11. D'autres données peuvent également faciliter l'appréciation de la performance d'un système d'exploitation. On peut, par exemple, envisager ce qui suit :

- a) la distribution des temps de service.
- b) Le pourcentage des tâches interactives. (on peut, à cet effet, définir une tâche interactive en disant qu'une telle tâche doit être complétée en un temps au plus égal à la valeur d'une tranche de temps standard.)
- c) La distribution du nombre d'opérations d'E/S disque effectuées par minute.



Cette liste est loin d'être exhaustive. Elle montre néanmoins que le problème qui consiste à cerner la performance d'un système est compliqué et que la mesure de celle-ci ne l'est pas moins. Si on prend le cas d'un système MUMPS-11, hormis la surveillance de l'état du système adoptée en premier, rien de ce qui a été décrit ci-dessus n'est mesurable.

L'investigation empirique, en tant que méthode d'analyse d'un système informatique, ne permet pas la formulation de prédiction concernant le comportement du système.

Les résultats fournis par cette méthode d'analyse peuvent néanmoins faire apparaître des phénomènes insoupçonnés, se produisant dans le système et outre ce mérite, ces résultats peuvent également servir de données représentant la réalité pour des études utilisant les modèles mathématiques ou la simulation.

#### 2.2.2. Description des conditions initiales du travail.

La société STERIABEL a procédé à l'informatisation de la section "ARBITRAGE" de la Banque de Bruxelles. L'idée du travail consistait, au départ, à exposer le problème de l'évaluation des performances de l'installation Banque de Bruxelles.

Par suite d'un certain nombre de difficultés, c'est à la société Interactive Systems que le travail a pu réellement débiter et se poursuivre efficacement. Les données dont on a disposé pour entamer ce mémoire, sont rassemblées et décrites en appendice A. Les installations que l'on a ainsi abordées au cours de ce travail étaient toutes deux équipées d'une machine PDP-11 munie du software MUMPS-11.

La surveillance entamée à la Banque de Bruxelles était du type de celle décrite en 2.2.1. Les éléments de l'échantillonneur d'états du système peuvent être détectés à partir de la description d'un état de sortie décrit en appendice A.

Si les deux installations étaient équipées du même hardware, et du même software, elles n'étaient pas pour autant identiques et le premier travail réalisé chez Interactive Systems fut celui de l'adaptation du programme de surveillance utilisé à la Banque de Bruxelles. Ce programme fut en fait, réécrit dans la version 2 de MUMPS-11.

Une description de ce programme est réalisée au paragraphe suivant.

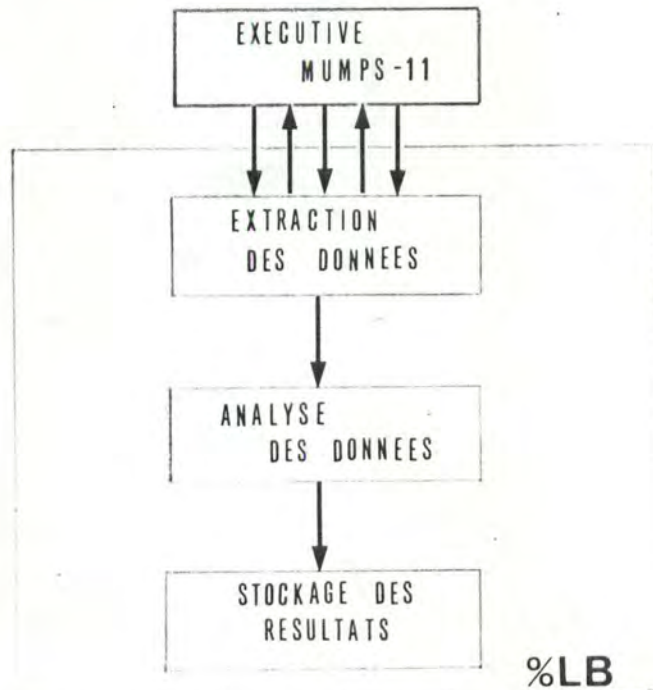


## 2.3. DESCRIPTION DU PROGRAMME DE SURVEILLANCE.

### 2.3.1. Description logique du programme.

Les références, dans la description qui va suivre, seront faites à la version adaptée du programme de monitoring (programme %LB).

La structure du programme peut être présentée comme suit :



a) L'extraction : elle consiste à placer :

1. dans la variable locale indicée JT, le contenu de JOBTAB. Les entrées programmes de JOBTAB sont placées dans JT(I), I=0:1:8. Les entrées files d'attente de JOBTAB sont placées dans JT(I), I=41:1:51.
2. Dans la variable locale indicée DT en DT(0), DT(2), DT(46), DT(62), les entrées #1, #3, #46, #63 de DEVTAB.
3. Dans les variables locales simples
  - T : le temps en seconde écoulée depuis minuit.
  - G : l'entrée 26 de SYSTAB "Garbage indicator".
  - WJ : l'entrée 154 de SYSTAB (nombre de jobs attendant le contrôle du processeur.)

b) L'analyse : elle consiste à construire la variable de type "string" E qui contiendra :

- le temps sous forme HH:MM (heure/minute)
- le nombre de jobs actifs calculé à partir de
- l'entrée 0 de JOBTAB
- l'état de chacune des 8 partitions

- le numéro de la partition occupant les devices  
1,3,46,63
- SYSTAB + 26
- SYSTAB + 154

c) Le stockage : il consiste à envoyer sur bande magnétique, le contenu de la variable type "string" E.

### 2.3.2. Description "pas à pas" du programme.

Convention : on va présenter le programme pas à pas. Chaque "step" sera, une première fois, présenté tel qu'il se trouve dans le programme et ensuite décomposé commande par commande.

1.10 S INT= $\wedge$ MON(0), ST=\$V(44), JT=\$V(ST), DT=\$V(ST+4), J1=JT+16, J2=JT+82

- S[ET] INT= $\wedge$ MON(0) : la variable INT contient la valeur de l'intervalle de monitoring qui se trouve à l'indice 0 de la variable globale MON.
- S[ET] ST=\$V[IEW](44) : la variable ST contient l'adresse de début de SYSTAB.
- S[ET] JT=\$V[IEW](ST) : la variable JT contient l'adresse de début de JOBTAB.
- S[ET] DT=\$V[IEW](ST+4) : la variable DT contient l'adresse de début de DEVTAB.
- S[ET] J1=JT+16 : J1 contient l'adresse de fin des entrées programmes de JOBTAB.
- S[ET] J2=JT+82 : J2 contient l'adresse de début des entrées files d'attente de JOBTAB.

1.15 S J3=JT+102, T1=ST+26, T2=ST+154, E="" A 47 P 5

- S[ET] J3=JT+102 : J3 contient l'adresse de fin des entrées files d'attente de JOBTAB.
- S[ET] T1=ST+26 : T1 contient l'adresse de l'entrée 26 de SYSTAB.
- S[ET] T2=ST+154 : T2 contient l'adresse de l'entrée 154 de SYSTAB.
- S[ET] E="" : la variable de type "string" E est destinée à contenir une ligne de monitoring. Elle est initialisée à blanc.
- A[SSIGN] 47 : on réserve la bande magnétique pour le programme de monitoring.
- P[RINT] 5 : on effectue le "rewind" de cette bande magnétique.



1.20 I  $\$V(65400)/100 \&.64 = .64$  T E @ "FIN\*", ! U 47 Q

- I[F]  $\$V[IEW](65400)/100 \&.64 = .64$
- T[YPE] E@ "FIN\*", !
- U[NASSIGN] 47
- Q[UIT]

Ces 4 commandes constituent le test d'arrêt du programme de monitoring. Si le bit 6 (masque .64) de la variable "SWITCH REGISTER" (adresse 65400) est positionné, on écrit un article "FIN\*" fin de fichier, on libère le device # 47 et on stoppe l'exécution du programme. On peut positionner le bit 6 de SWITCH REGISTER à partir de "Keyboard" du processeur.

1.30 H 0,0,0 S T = \$T, G = \$V(T1), WJ = \$V(T2) F I = 0,2,46,62  
S DT(I) = \$V(DT+I)

- H[ANG] 0,0,0 : cette commande est destinée à forcer le job en "LOW PRIORITY WAIT QUEUE" de manière à ce que l'on puisse entamer l'extraction des informations relatives à l'état du système avec la plus grande tranche temps. (voir à ce propos le "system queuing directive" CLKREQ en appendice).
- S[ET] T = \$T
- S[ET] G = \$V[IEW](T1)
- S[ET] WJ = \$V[IEW](T2)
- F[OR] I = 0,2,46,62 S[ET] DT(I) = \$V[IEW](DT+I)

Ces 4 commandes réalisent l'extraction de :

- le temps écoulé depuis minuit.
- l'entrée 26 de SYSTAB.
- l'entrée 154 de SYSTAB.
- les 4 entrées de DEVTAB.

1.40 F I = JT:2:J1, J2:2:J3 S JT(I-JT/2) = \$V(I)

- F[OR] I = JT:2:J1, J2:2:J3 S[ET] JT(I-JT/2) = \$V[IEW](I)

Cette commande réalise l'extraction du contenu des entrées programmes et files d'attente de JOBTAB.

1.50 S N1 = JT(0)/25600, N2 = JT(0)/100, NJ = N1 - (N2 & 2.55) \* 100  
S: NJ? 1D NJ = 0 @ NJ

- S[ET] N1 = JT(0)/25600 : N1 reçoit le contenu du high byte de l'entrée 0 de JOBTAB qui représente le nombre maximum de partitions du système.
- S[ET] N2 = JT(0)/100 : N2 est destiné à contenir le low byte de l'entrée 0 de JOBTAB qui représente le nombre courant de partitions disponibles.



Il est à remarquer que ce low byte sera seulement extrait à l'assignation suivante par l'opération :  
 $N2 \& 2.55 * 100$

- S[ET] NJ=N1-(N2 2.55)\*100 : NJ contient le nombre maximum de partitions disponibles, diminué du nombre courant de partitions disponibles, ce qui donne le nombre de jobs actifs dans le système. Cette valeur doit en principe être égale à JW augmenté d'une unité.
- S:NJ?1D NJ=0 NJ : par cette instruction, on fait la concaténation de 0 avec NJ si NJ se compose d'un seul digit.

1.60 D 2 S E = E @ H @ " : " @ M @ " \* " @ N J @ " \* " F I = 1 : 1 : N 1 \* 100 D 4  
 1.70 G 3

- D[0] 2 : correspond à l'exécution de la procédure qui transforme le temps fourni en nombre de secondes écoulées depuis minuit sous la forme Heure/minute.
- S[ET] E=E H " : " M " \* " N J " \* " : on commence à remplir le string E avec le temps et le nombre de jobs actifs.
- F[OR] I=1:1:N1\*100 D 4 : par cette boucle, on va analyser chaque entrée programme de JOBTAB. I représente donc le numéro de la partition que l'on veut analyser.
- G[0] 3 : branchement à la procédure 3 à la fin de la procédure 1.

2.10 S M = \$I(T/60), N = \$E("AP", \$I(M/720)+1) @ "M", M = -\$I(M/720)\*720+  
 2.20 S H = \$I(M/60), M = -H\*60+M S: N="PM" H=H+12 S: H?1D H=0 @ H  
 S: M?1D M=0 @ M

Ces deux steps constituent la procédure de mise en forme du temps.

	\$T=40000	\$T=50000
• S[ET] M=\$I[NTEGER](T/60)	M=666	M=833
• S[ET] N=\$E[XTRHCT]("A", \$I[NTEGER](M/720)+1) @ "M"	N=AM	N=PM
• S[ET] M=\$I[NTEGER](M/720)*720+M	M=666	M=113
• S[ET] M=\$I[NTEGER](M/60)	H=11	H=1
• S[ET] M=-H*60+M	M=6	M=53
• S:N="PM" H=H+12	/	H=13
• S:H?1D H=0 @ H	/	/
• S:M?1D M=0 @ M	M=06	/
	↑	↑
	11:06	13:53



4.10 G 4.3:JT(I)/25600&1.28 D 4.2 S:SW E=E@ "32\*" Q

• G[0] 4.3:JT(I)/25600&1.28 : on branche en 4.3 si le HANG bit (appendice) d'une entrée programme est positionné. Sinon :

• D[0] 4.2

• S:SW E=E "32\*"Q[UIT] : on a trouvé le job dans aucune file d'attente : on lui attribue l'état 32.

4.20 F J= 41:1:51 D G I Y=I S E=E@J@ "\*" Q

Pour le job (partition) n°I, on va parcourir les files d'attente 41 - 51 pour déterminer celle dans laquelle il est chaîné (procédure 6). Lorsqu'on a trouvé, on place le n° de la file d'attente, dans laquelle il se trouvait, dans le string E et on stoppe.

4.30 S X=JT(I)/25600\*100 S: X=192 X=28 S:X=196 X=29  
S:X=200 X=30

4.40 S E=E@X@ "\*" Q

Lorsque dans une entrée programme, on a constaté que le HANG BIT était positionné, on cherche alors si le contenu du HIGH BYTE correspond à un état "Hung terminal", Hung Deftape" ou "Hung Magtape" puis on place l'état trouvé dans le string E et on stoppe.

6.10 S K=JT(J)/100&2.55\*50, SW=1

On place dans K l'adresse qui pointe sur le début de la file d'attente J.

6.20 F D=.01:.01 W K&(D<=N1) S Y=K, K=JT(K)/100&2.55\*50 I Y=I  
S SW=0 Q

On remonte maintenant la liste des jobs chaînés dans la file d'attente J. On s'arrête dès que l'on a trouvé la file d'attente correspondant à la partition fixée où dès que l'on atteint la fin de la file d'attente.

3.05 DS S G=G/100&2.55\*100 I G=0 S G="00" G 3.2

• D[0] 5 : dans la procédure 5, on détermine l'état des quatre devices envisagés.

• S[ET] G=G/100 2.55\*100 I[F] G=0 S[ET] G="00" G[0] 3.2  
On analyse le contenu de l'entrée 24 de SYSTAB (garbage indicator). Si cette entrée contient 0, on placera "00" dans le string E sinon (step 3.10) on place dans E les caractères "01".

3.10 S G="01"

3.20 SWJ=WJ/100&2.55\*100 S:WJ?10 WJ.0@WJ T E@G@"@"@WJ@"\*",

On analyse dans le step 3.20 le contenu de l'entrée 154 de SYSTAB (Number of job waiting to run). Cette analyse termine le remplissage du string E avec une ligne de monitoring. Ce string est alors envoyé sur bande magnétique par la commande T[YPE].



3.30  $SE = "$ ",  $N = INT - (\$T - T) \quad \# \quad N \in 1,2$

•  $S[ET] \quad N = INT - (\$T - T)$

•  $H[ANG] \quad N$

on Place dans la variable N le temps pendant lequel on veut suspendre le programme. Ce temps est égal à l'intervalle de monitoring diminué du temps d'exécution du programme.

5.10  $S \quad Q(46) = DT(46) / 100 \times 2.55 \times 50 \quad FI = 0,2,62 \quad S \quad Q(I) = DT(I) / 25600 \times 50$

On place en  $Q(0), Q(2), Q(46), Q(62)$  le numéro du job (ou partition) qui utilise le ou les devices 1,3,46,63.

5.20  $FI = 0,2,46,62 \quad S:Q(I) ? 10 \quad Q(I) = 0 @ Q(I) \quad SE = E @ Q(I) @ "$

On fixe l'état de chaque device dans 2 caractères et on le place dans le string E.

#### Remarque :

- 1° Etant donné que la fonction  $\$V[IEW]$  est l'élément clé de ce programme de monitoring, on décrira, dans cette première remarque, le fonctionnement précis de cette fonction.

$\$V[IEW]$  (valeur numérique d'une expression).

La fonction  $\$V[IEW]$  renvoie le contenu d'une zone de mémoire adressé par la valeur de l'argument de la fonction. Cette fonction renvoie toujours le contenu d'un mot mémoire. L'argument de la fonction doit être un nombre pair. Si c'est un nombre impair, une unité est soustraite automatiquement. Seule la portion entière de l'argument est prise en considération. Exemple :

$\$V(3) \rightarrow$  contenu du mot mémoire d'adresse  $2_{10}$ .

$\$V(25.79) \rightarrow$  contenu du mot mémoire d'adresse  $24_{10}$ .

- 2° Représentation MUMPS-11 des données numériques.

Mumps-11 crée des nombres en format MUMPS ayant un profil binaire correspondant à 100 fois le contenu initial du nombre.

Par exemple, si on fait  $S \quad A=1$ , le profil binaire du contenu de la variable A correspond à 100.

Donc :  $A \neq 00000001$

mais bien :  $A = 01100100 \equiv 100_{10}$

On peut vérifier que l'on a bien ce profil en testant les bits 2,5,6. Il est toutefois évident que les masques à utiliser sont à diviser par 100.

Ainsi, pour tester le bit 2 de la variable A, il faut utiliser  $2^2/100 = 0.04$ . On peut ainsi vérifier que :  $I \quad A \&.04 \rightarrow \text{TRUE}$



```

I A&.32 → true
I A&.64 → true
I A&1.00 → true
(1.00 = 0.04+0.32+0.64)

```

Si maintenant, on sait que le résultat renvoyé par la fonction \$V[IEW] est 100 fois plus grand que le contenu réel, on comprend les opérations nécessaires, par exemple, pour extraire le "low byte" d'une entrée d'une table du système extraite par l'intermédiaire de \$V[IEW].

Ex.: extraire le low byte de l'entrée 0 de JOBTAB.

extraction : S JT(0)=\$V(\$V(44))

low byte : S N2=JT(0)/100&2.55\*100

représentation du contenu réel

ET logique de ce mot avec un masque correspondant à 7 bits positionnés.

on multiplie par 100 pour re-devenir compatible avec la représentation interne MUMPS-11.

## 2.4. Présentation d'outils complémentaires d'analyse.

Une fois le programme de surveillance adapté, on a écrit un certain nombre de programmes destinés à réaliser des récapitulatifs tels que ceux présentés dans l'appendice A. On a ainsi écrit :

- a) le programme LB4 qui fournit un récapitulatif par partition de l'échantillon de surveillance.
- b) Le programme LB3 qui délivre un récapitulatif global de l'échantillon de surveillance.
- c) Le programme LB2 qui offre une sortie sur imprimante de l'échantillon de surveillance.

En plus de ces résultats, on a écrit un programme LBO qui construit un tableau du nombre d'occurrences de chaque état ou file d'attente présent dans un instantané du système. Quant au programme LB1, il réalise l'impression des distributions du nombre d'occurrences.

Pour compléter l'ensemble des résultats reçu de la première installation, on a fait l'analyse selon l'optique des programmes LBO et LB1, de la sortie sur imprimante du monitoring (voir A.2. de l'appendice A). Le résultat de cette analyse est présenté dans TAB - 2.4.1., TAB - 2.4.2. et TAB - 2.4.3. où on constate une proportion non négligeable de jobs dans l'état "en transition" et une population inattendue (au vu de la description du système des files d'attente) dans les files d'attente 42 et 43.

D'autre part, on a aussi observé au cours de l'analyse de la sortie imprimante :

- a) un non respect de la fréquence des "snapshots" du système.
- b) deux instantanés du système qui étaient incohérents. Ces deux instantanés sont les suivants :

15:47/06/51/28/44/44/28/44/44/43/41/44/28/44/51/51/10/00/00/00

8 partitions disponibles

16:59/07/51/28/44/44/43/44/28/48/41/50/44/28/51/51/00/00/00/00

7 partitions disponibles

Ces observations vont orienter le travail de la manière suivante:

- a) on va approfondir au maximum le problème de l'état "en transition".
- b) on essaiera d'apporter des améliorations au programme de surveillance dont on s'est servi pour réaliser la surveillance de l'installation Banque de Bruxelles.



## 2.5. CONCLUSIONS.

2.5.1. En consultant les brochures MUMPS-11, on peut constater qu'il existe un programme utilitaire qui effectue des "snapshots" sur le système. C'est le programme SS dont un exemple de sortie est présenté en fin de travail. Cela étant, on peut alors constater que l'auteur du programme de surveillance s'est borné à écrire la commande qui suspendait le travail pour une certain délai et la commande qui le branchait sur lui-même pour une nouvelle exécution.

Dans ces conditions, on peut remarquer dès l'abord que pour le programme utilitaire qui est destiné à effectuer des instantanés du système isolé dans le temps, il est indispensable que l'analyse suive immédiatement l'extraction des informations. Il n'en n'est pas de même pour le programme de surveillance ! A priori, on peut penser que cette analyse "on line" des informations va coûter du temps inutile. En conséquence, le but poursuivi par celui qui a créé le programme utilitaire est nettement différent du but poursuivi par l'auteur du programme de surveillance et, si le procédé conçu pour l'utilitaire semble raisonnable, l'auteur du programme de surveillance semble avoir commis une faute de choix en utilisant un programme dans un but pour lequel il n'a pas été conçu.

2.5.2. En consultant la documentation relative au programme utilitaire SS, on peut constater que le manuel reste extrêmement évasif lorsqu'il s'agit d'expliquer que le programme SS n'est pas à même de prélever une image unique du système. Cette inaptitude à pouvoir fournir une image unique de système est illustrée par le fait qu'un état "EN TRANSITION" ait été créé.

Cet état "EN TRANSITION" ne correspond pas à un des états dans lequel un job peut se trouver vis-à-vis du système; il s'agit plutôt d'une étiquette que l'on appose à une partition qui, au cours de la phase d'analyse du programme SS, n'a été trouvé dans aucun état ou aucune file d'attente.

Cet état "en transition" donne à la fois une mesure du caractère dynamique d'une recharge du système ainsi qu'une mesure du degré de confiance que l'on peut avoir en le programme de surveillance.

2.5.3. L'idée qui a conduit à l'existence des files d'attente "SHORT QUEUE" et "DISK I/O BOUND QUEUE" consistait à vouloir privilégier le programme qui effectuait un I/O disque par rapport à tous les autres programmes d'application dans le système.

Cette idée impliquait également le fait qu'un seul programme pouvait se trouver dans cette paire de files d'attente. On peut tester cette hypothèse par l'intermédiaire de la distribution des occurrences de l'état 42 et 43.

Ces distributions ne doivent en effet contenir que les occurrences 0 et 1, ce qui ne semble pas être le cas sur la fig. 2.4.2.



FILE DATTENTE 28 [I/O hung TERMINAL]

```

0 1 *
1 27 ****
2 51 *****
3 99 *****
4 103 *****
5 109 *****
6 49 *****
7 1 *

```

FILE DATTENTE 29 [I/O hung DECTAPE]

```

0 440 *****

```

FILE DATTENTE 30 [I/O hung MAGTAPE]

```

0 439 *****
1 1 *

```

FILE DATTENTE 32 [IN TRANSITION]

```

0 334 *****
1 63 *****
2 21 *****
3 2 *

```

FILE DATTENTE 41 [RUN QUEUE]

```

1 440 *****

```



## FILE D'ATTENTE 42 [SHORT QUEUE]

0	384	*****
1	41	*****
2	6	*
3	4	*
4	3	*

## FILE D'ATTENTE 43 [DISK I/O BOUND QUEUE]

0	225	*****
1	195	*****
2	13	**
3	6	*
4	1	*

FILE D'ATTENTE 44 [DISK RESOURCE BOUND  
QUEUE]

0	250	*****
1	60	*****
2	44	*****
3	20	***
4	7	*
5	1	*

FILE D'ATTENTE 45 [RING BUFFER RESOURCE  
BOUND QUEUE]

0	440	*****
---	-----	-------

FILE D'ATTENTE 46 [256-WORD BUFFER  
RESOURCE BOUND QUEUE]

0	440	*****
---	-----	-------

## FILE D'ATTENTE 47 [HIGH WAIT QUEUE]

0	401	*****
1	39	*****

## FILE D'ATTENTE 48 [MEDIUM WAIT QUEUE]

0	219	*****
1	112	*****
2	9	**

## FILE D'ATTENTE 49 [LOW WAIT QUEUE]

0	388	*****
1	83	*****
2	45	*****
3	16	***
4	6	*
5	1	*

## FILE D'ATTENTE 50 [CLOCK QUEUE]

0	313	*****
1	120	*****
2	7	*

FILE D'ATTENTE 51 [PARTITION AVAILABLE  
QUEUE]

3	4	*
4	24	****
5	94	*****
6	125	*****
7	128	*****
8	58	*****
9	7	*



### CH A P I T R E   I I I   :

-----

#### LIMITES ET RECHERCHE D'AMELIORATIONS

.....

#### DU PROGRAMME DE SURVEILLANCE

.....

- 3.1. Mesure du temps d'exécution de la procédure extractive.
- 3.2. Mesure du temps d'exécution du programme de surveillance.
- 3.3. Recherche d'améliorations.
- 3.4. Surveillance d'une même charge par deux versions du programme.
- 3.5. Etude des variations du temps d'exécution.
- 3.6. Résultats parallèles.

### 3.1. MESURE DU TEMPS D'EXECUTION DE LA PROCEDURE EXTRACTIVE.

Les expériences présentées dans ce paragraphe sont toutes relatives à des mesures de temps d'exécution. Ces mesures ont toutes été réalisées dans la condition de seul utilisateur de la machine.

3.1.1. La première mesure effectuée a été celle du temps d'exécution de la procédure d'extraction des informations relatives à l'état du système. Cette mesure a été produite par le programme % B Ø. Le résultat fourni par ce programme est assez surprenant : le temps moyen d'exécution de la procédure d'extraction est de 456.20 millisecondes, soit 22.60 ticks (voir TAB-3.1.1).

Ce second chiffre (22.6 ticks) est à mettre en rapport avec la valeur d'une grande tranche de temps: 15 ticks. Si la valeur de 22.6 ticks est certainement entachée d'un temps "d'overhead" du système, il n'est cependant pas raisonnable d'admettre qu'il puisse être d'au moins 7 ticks (soit 140 millisecondes) et dès lors, on peut se demander si la façon de procéder, pour extraire les informations du système, n'est pas inadéquate pour relever les informations relatives à un état unique du système.

3.1.2. Ce résultat surprenant étant acquis, on a voulu voir comment se distribuait le temps parmi les diverses instructions qui constituent la procédure d'extraction, afin de pouvoir effectuer une vérification du premier résultat obtenu.

Le programme % M M a été écrit à cet effet. Dans ce programme, on a mesuré le temps d'exécution des instructions suivantes :

3.1.2.1.  $ST = \$T$ ,  $G = \$V(T3)$ ,  $WJ = \$V(T4)$

Les résultats sont présentés dans TAB-3.1.2 et la valeur moyenne obtenue est de 18.32 millisecondes. Toutefois, si on prend la partie 2 du programme % M M on voit que l'on a mesuré en plus le temps d'exécution de la commande FOR. Ce temps a été mesuré par le programme MEG et est en moyenne de 3.5 millisecondes.

Dès lors, le temps moyen de l'instruction envisagée est de 14.82 millisecondes.

3.1.2.2.  $FK = 0, 2, 4, 6, 8$   $SDT(K) = \$V(DT + K)$

Les résultats de la mesure sont présentés dans TAB-3.1.3. La valeur moyenne obtenue, le temps de la commande FOR soustrait, est de 43.91 millisecondes.



### 3.1.2.3. $F \ K = J_T : 2 : J_1, J_2 : 2 : J_3 \quad S \ J_T(K - J_T/2) = \$V(K)$

Les résultats sont présentés dans TAB-3.1.4  
et la valeur moyenne obtenue est de 359.96.

En sommant les résultats des points 1,2,3, on obtient un temps moyen total de 418.69 millisecondes. Comparé au résultat de la première mesure (456.2 - 3.5 = 452.5) on obtient une différence de 33.51 millisecondes soit 1.67 ticks ce qui rend toutes les mesures compatibles en tenant compte du temps de mesure et d'overhead du système.

On a ensuite repris les instructions 2 et 3 pour lesquels on a effectué le même travail. Les résultats sont les suivants :

### 3.1.2.4. $F \ K = 0, 2, 46, 62 ;$

Les résultats sont présentés dans TAB-3.1.5  
et le temps moyen obtenu est de 15.75 - 3.5 = 12.25 millisecondes.

### 3.1.2.5. $F \ K = J_T : 2 : J_1, J_2 : 2 : J_3 ;$

Les résultats sont présentés dans TAB-3.1.6;  
le temps moyen obtenu est de 59.24 - 3.5 = 55.74 millisecondes.

### 3.1.2.6. $S \ DT(K) = \$V(DT + K)$

Les résultats sont présentés dans TAB-3.1.7;  
le temps moyen obtenu est de 16.43 - 3.5 = 12.93 millisecondes.

### 3.1.2.7. $S \ J_T(K - J_T/2) = \$V(K)$

Les résultats sont présentés dans TAB-3.1.8;  
le temps moyen obtenu est de 18.68 - 3.5 = 15.18 millisecondes.

Le temps de l'instruction 2 peut être reconstitué en calculant  $12.25 + 4 * 12.93 = 63.97$  qui comparé au résultat global de 43.91 présente une différence de 20.06 millisecondes, soit 1 tick. En effectuant le même travail pour l'instruction 3, on obtient un temps global reconstitué de 435.24 millisecondes, qui présente une différence de 75.28 millisecondes, soit 3.76 ticks par rapport à la valeur de la mesure du temps global. Remarquons tout de même que ces différences (1 et 3.76 ticks) sont raisonnables si l'on songe que l'observation du temps dans le système MUMPS-11 n'est exacte qu'à un tick près.

3.1.3. On a enfin voulu mesurer que coûtait en temps l'exécution de la fonction  $\$V[IEW]$  sans laquelle l'observation de l'état du système ne serait



pas possible. On a utilisé pour cela, le programme MEG. Les résultats obtenus sont les suivants :

- 3.1.3.1. Tout d'abord, on présente dans TAB-3.1.9, les résultats obtenus pour la mesure du temps d'exécution de la commande FOR, le résultat moyen étant de 3.48 millisecondes, soit 0.17 ticks. Pour effectuer cette mesure, on a programmé un Q UIT en 1.99.
- 3.1.3.2. On a ensuite mesuré le temps d'exécution de la \$V [IEW] fonction avec argument constant. On a programmé pour cela un S X = \$V(44) en 1.99. Les résultats sont présentés en TAB- 3.1.10 et le résultat moyen obtenu est de 7 - 3.5 = 3.5 millisecondes soit 0.17 tick.
- 3.1.3.3. Le même travail a été fait pour la \$V [IEW] fonction avec argument variable. La commande en 1.99 est devenu S J = 44, X = \$V(J). Les résultats sont présentés dans TAB- 3.1.11 et la valeur moyenne obtenue est de 9.94 - 3.5 = 6.44 millisecondes, soit 0.32 tick.
- 3.1.4. La mesure du temps d'exécution d'une instruction (commande et/ou fonction) n'apparaît pas comme un travail trivial comme on pourrait le penser à priori. Tout d'abord, dans un contexte de temps partagé, il faut se rendre compte que toute mesure est entachée d'un certain temps "d'overhead" du système. Ensuite la manière de concevoir et de réaliser une mesure influencent également le résultat.
- A cet effet, on va montrer deux choses.
- 3.1.4.1. On a conçu un autre programme (%L1) destiné à effectuer le même travail que le programme MEG. On montre ci-dessous une comparaison des résultats obtenus dans la mesure du temps d'exécution de la fonction \$V [IEW] à argument constant.

A	MEG		%L1	
	B	C	B	C
100	1	10	0	0
200	1	5	2	10
300	3	10	3	10
400	3	7.5	3	7.5
500	3	6	4	8
600	4	6.66	5	8.33
700	4	5.71	6	8.57
800	5	6.25	7	8.75
900	6	6.66	7	7.77



1000	7	7	8	8
1100	7	6.36	9	8.18
1200	8	6.66	10	8.33
1300	9	6.92	11	8.46
1400	10	7.14	12	8.57
1500	10	6.66	13	8.66
2000	14	7	17	8.50
3000	21	7	25	8.33
4000	28	7	34	8.50
5000	35	7	43	8.60

**Légende :**

La colonne référencée A contient la répétition.

Les colonnes référencées B contiennent le temps total observé (en sec.).

Les colonnes référencées C contiennent le temps d'exécution de la fonction (en milliseconde.).

Si on effectue le calcul de la valeur moyenne à partir de la répétition 1000, on obtient, dans le cas MEG, un temps moyen de 6.87 millisecondes et, dans le cas %L1, un temps moyen de 8.41 millisecondes.

**3.1.4.2. On a effectué le calcul du temps d'exécution de la FOR Command de la manière suivante :**

1.20 .... F I = 1:1:TTC D 1.99

.....  
1.99 Q

On a ensuite essayé de refaire la même mesure en utilisant le programme %L1 pour des instructions qui pouvaient correspondre à ne rien faire.

On a, pour cela, tenté :

a) H Ø → 6.60 millisecondes

b) A Ø → 5.70 millisecondes

c) ; commentaire → 4.60 millisecondes.

**3.1.5. CONCLUSION.**

La fonction \$V [IEW] n'apparaît pas comme étant un moyen adéquat et surtout rapide pour extraire des informations du système. Il faut en plus constater que c'est le seul dont on dispose.

PRESENTATION DES RESULTATS DE MESURES.

A : répétition de l'instruction à mesurer.

B : temps total observé (sec.).

C : temps calculé pour une exécution (millisec.).

A	B	C
100	45	450
200	92	460
300	137	456.66
400	183	457.50
500	228	456
1000	457	457

TAB - 3.1.1.

A	B	C
1000	18	18
2000	37	18.50
3000	55	18.33
4000	73	18.25
5000	92	18.40
10000	185	18.50

TAB - 3.1.2.

A	B	C
600	24	48
600	28	46.66
700	33	47.14
800	38	47.50
900	43	47.77
1000	47	47
1100	53	48.18
1200	57	47.50
1300	61	46.92
1400	67	47.85
2000	94	47

TAB - 3.1.3.



A	B	C
500	180	360.00
600	218	363.33
700	255	354.28
800	291	363.75
900	328	364.44
1000	365	365.00

TAB - 3.1.4.

A	B	C
500	8	16
600	10	16.66
700	11	15.71
800	12	15
900	14	15.55
1000	16	16
1100	17	15.45
1200	19	15.83
1300	20	15.38
1400	22	15.71
2000	32	16

TAB - 3.1.5.

A	B	C
500	30	60
600	35	58.33
700	41	58.57
800	48	60
900	53	58.88
1000	59	59
1100	66	60
1200	71	59.16
1300	77	59.23
1400	83	59.28
1500	119	59.50

TAB - 3.1.6.

A	B	C
500	8	16
600	10	16.66
700	12	17.14
800	13	16.25
900	15	16.66
1000	16	16
1100	18	16.36
1200	20	16.66
1300	21	16.15
1400	23	16.42
2000	33	16.50

TAB - 3.1.7.

A	B	C
500	9	18.00
600	12	20.00
700	13	18.57
800	15	18.75
900	16	17.77
1000	19	19.00
1100	21	19.09
1200	22	18.33
1300	24	18.46
1400	26	18.57
2000	38	19.00

TAB - 3.1.8.

A	B	C
1000	4	4
2000	7	3.50
3000	10	3.33
4000	13	3.25
5000	17	3.40
10000	34	3.40

TAB - 3.1.9.



A	B	C
1000	7	7
2000	14	7
3000	21	7
4000	28	7
5000	35	7
6000	42	7
7000	49	7
8000	56	7
9000	63	7
10000	70	7

TAB - 3.1.10.

A	B	C
1000	10	10
2000	20	10
3000	29	9.66
4000	40	10
5000	50	10
6000	59	9.83
7000	70	10
8000	80	10
9000	90	10
10000	100	10

TAB - 3.1.11.

### 3.2. MESURE DU TEMPS D'EXECUTION DU PROGRAMME DE SURVEILLANCE.

Les expériences présentées dans ce paragraphe concernent la mesure du temps d'exécution du programme de surveillance. Parallèlement à cette mesure, on a également essayé de voir comment se distribuait le temps pour réaliser une exécution du programme.

Pour effectuer ces mesures, on a utilisé les informations contenues dans 3 entrées de SYSTAB (voir appendice).

Lors d'une observation du temps, quatre informations sont prélevées et stockées dans des variables locales de la partition :

- a) l'information contenue dans le \$T fonction qui représente le temps en seconde écoulé depuis minuit.
- b) SYSTAB + 28 qui contient le nombre de ticks restant dans la seconde courante (une seconde = 50 ticks).
- c) SYSTAB + 30 qui contient le nombre de ticks restant dans la tranche de temps courante.
- d) SYSTAB + 31 qui contient le nombre de ticks attribués à la tranche de temps courante.

Les informations sont extraites par l'intermédiaire de la fonction \$V[IEW] et la procédure de prélèvement est dans l'ordre où les variables ont été définies :

- a) S [ET] T(L)=\$T
- b) S [ET] A(L)=\$V(\$V(44)+28)/100&2.55\*100
- c) S [ET] B(L)=\$V(\$V(44)+30)/100&2.55\*100
- d) S [ET] C(L)=\$V(\$V(44)+30)/25600\*100

Quant à la séquence d'appel, elle est la suivante :

S [ET] L=L+1 D (numéro attribué à la procédure de prélèvement.)

3.2.1. La première mesure réalisée fut celle du temps d'exécution de la procédure d'observation du temps. Cette mesure a été effectuée par le programme %E1. Les résultats de la mesure sont rassemblés dans TAB - 3.2.1. Le temps moyen d'exécution de la procédure d'observation du temps est de 31.86 millisecondes, soit 1.59 ticks.

3.2.2. On a ensuite inséré cette procédure de prélèvement du temps à divers endroits du programme de surveillance afin de mesurer le temps d'exécution de ce programme. Les prélèvements du temps ont été réalisés de manière à obtenir une mesure du temps d'analyse et une mesure du temps d'extraction. Il est à remarquer que cette mesure a déjà été faite dans le paragraphe précédent. Il est toutefois intéressant de reproduire cette mesure dans un autre contexte afin de vérifier à nouveau la cohérence des résultats antérieurs.

3.2.2.1. On s'est d'abord placé dans les conditions de la première installation où il fallait observer l'état de 14 partitions. (La modi-



fication à apporter dans le programme de surveillance se situe au niveau du step 1.40 où, dans la For loop, la première suite d'indices se termine en  $JT + 30$ ).

L'expérience s'est déroulée alors que l'on était seul à utiliser la machine à ce moment. Le programme qui a effectué ces mesures est le programme %E2. Le résultat de cette expérience est présenté dans TAB - 3.2.2.

#### 3.2.2.2. Description des résultats.

Chaque groupe de 6 lignes du tableau concerne une exécution du programme de surveillance. Les endroits du programme où le temps a été observé sont les suivants :

ligne 1 : step 1.25 ; avant la commande  $H\emptyset, \emptyset, \emptyset$   
 ligne 2 : step 1.30 ; après la commande  $H\emptyset, \emptyset, \emptyset$ .  
 C'est aussi le début de l'extraction des données relatives à l'état du système.

ligne 3 : step 1.35 ; pendant l'extraction.

ligne 4 : step 1.45 ; fin de l'extraction qui est aussi le début de l'analyse des données recueillies.

ligne 5 : step 3.3 ; fin d'analyse.

ligne 6 : step 3.3 ; après la suspension du temps d'intervalle de monitoring.

La signification du contenu de chacune des 6 colonnes du tableau est la suivante :

colonne 1 : valeur de \$T fonction.

colonne 2 : valeur de SYSTAB + 28

colonne 3 : valeur de SYSTAB + 30

colonne 4 : valeur du SYSTAB + 31

colonne 5 : temps d'extraction obtenu en soustrayant le contenu de la colonne 2 des lignes 4 et 2.

colonne 6 : temps d'analyse obtenu en soustrayant le contenu des colonnes 1 et 2, qui fournit le temps au tick près des lignes 5 et 4.

A l'aide de cette expérience, on observe ainsi un temps moyen d'extraction de 33.5 ticks, soit 0.670 seconde. En fait de ce temps d'extraction, il faut soustraire trois fois le temps d'observation du temps, soit  $0.03186 * 3 = 0.095$  sec., ce qui donne pour le temps d'extraction une valeur de 0.576 sec.

Quant au temps moyen d'analyse des informations prélevées, on le trouve agal à 196.75 ticks, soit 3.935 sec.

#### 3.2.2.3. On a refait la même mesure en se plaçant dans les conditions de la seconde installation (8 partitions à observer/Step 1.40: la première

suite d'indices se termine en JT+16).  
Les résultats sont présentés dans TAB - 3.2.3.  
Le temps moyen d'extraction calculé est de  
27.5 ticks soit 0.550 sec. A ce temps, il faut  
soustraire également 0.095sec. pour observer  
le temps, d'où un temps d'extraction de 0.455  
seconde . Comparé à la valeur obtenue au para-  
graphe précédent (0.456 sec.), on obtient une  
excellente cohérence dans les mesures. En fait,  
il faut comprendre que la précision des mesures  
(+ 2 ticks) a joué dans le même sens dans les  
deux mesures séparées que l'on a fait du même  
phénomène.  
Le temps d'analyse a, à nouveau, été observé  
à 195.5 ticks, soit 3.91 secondes.



PRESENTATION DES RESULTATS DE MESURES.

A : répétition de la { commande à mesurer  
                                  { procédure

B : temps total observé (en secondes).

C : temps calculé d'exécution de la procédure (millisecondes).

A	B	C
1000	32	32
2000	63	31.50
3000	95	31.66
4000	128	32
5000	160	32
10000	320	32

TAB - 3.2.1.

\$T	S28	S30	S31	$\Delta 1$	$\Delta 2$
49487	47	0	5	32	196
49487	43	13	15		
49487	37	7	15		
49487	11	0	15		
49491	15	0	15		
49498	49	4	5		
49498	46	0	5	32	196
49498	42	14	15		
49498	36	7	15		
49498	10	0	15		
49502	14	0	15		
49509	49	3	5		
509	45	0	5	33	196
509	42	16	15		
509	35	6	15		
509	9	0	15		
513	13	0	15		
520	49	3	5		
20	45	0	5	33	197
20	41	12	15		
20	34	5	15		
20	8	0	15		
24	11	0	15		
31	49	3	5		
31	45	0	5	34	197
31	41	13	15		
31	34	6	15		
31	7	0	15		
35	10	0	15		
42	49	3	5		
42	45	0	5	34	197
42	40	12	15		
42	33	5	15		
42	6	0	15		
46	9	0	15		
53	49	3	5		
53	44	0	5	35	197
53	40	13	15		
53	32	5	15		
53	5	0	15		
57	8	0	15		
64	49	3	5		
64	44	0	5	35	198
64	39	12	15		
64	31	4	15		
64	4	0	15		
68	6	0	15		
75	49	3	5		

TAB - 3.2.2.



\$T	S28	S30	S31	$\Delta 1$	$\Delta 2$
49872	46	0	5	26	195
72	43	13	15		
72	37	7	15		
72	17	0	15		
76	22	0	15		
83	49	4	5		
83	46	0	5	26	195
83	42	14	15		
83	36	7	15		
83	16	0	15		
87	21	0	15		
94	49	3	5		
94	45	0	5	27	195
94	42	13	15		
94	35	6	15		
94	15	0	15		
98	20	0	15		
905	49	3	5		
05	45	0	5	27	195
05	41	12	15		
05	34	5	15		
05	14	0	15		
09	19	0	15		
16	49	3	5		
16	45	0	5	28	196
16	41	13	15		
16	34	6	15		
16	13	0	15		
20	17	0	15		
27	49	3	5		
27	45	0	5	28	196
27	40	12	15		
27	33	5	15		
27	12	0	15		
31	16	0	15		
38	49	3	5		
38	44	0	5	29	196
38	40	13	15		
38	32	5	15		
38	11	0	15		
42	15	0	15		
49	49	3	5		
49	44	0	5	29	196
49	39	12	15		
49	31	4	15		
49	10	0	15		
53	14	0	15		
60	49	3	5		

TAB - 3.2.3.

### 3.3. RECHERCHE D'AMELIORATION.

Les expériences présentées dans ce paragraphe sont relatives à des améliorations que l'on a imaginées, à la fois, pour augmenter la performance du programme de surveillance et pour approfondir la connaissance du système MUMPS-11.

3.3.1. Ainsi qu'il a déjà été signalé dans un paragraphe précédent, s'il est indispensable d'effectuer immédiatement l'analyse des données extraites des tables du système lorsque l'on réalise des "snapshots" isolés dans le temps, cette analyse "on line" des informations apparaît toutefois superflue dès que l'on répète l'opération "snapshot" sur le système.

Aussi, a-t-on voulu réduire le programme de surveillance à l'extraction seule des informations relatives à l'état du système.

On a écrit à cet effet le programme %E3. Dans ce programme, on a envisagé l'extraction de la manière suivante :

- a) on va placer les informations du système sur disque dans la variable globale  $\uparrow$ MON.
- b) les données seront placées dans une variable de type "string" au fur et à mesure de leur extraction. On utilisera deux strings pour cela. De cette manière, également, on limitera à deux le nombre d'accès logiques au disque à chaque snapshot du système.

On a alors réalisé une mesure du temps d'exécution de ce programme. La procédure d'observation du temps a été la même que celle utilisée et décrite plus haut. Les résultats de cette mesure sont présentés dans TAB - 3.3.1.

Chaque groupe de 6 lignes du tableau concerne une exécution du programme %E3. Les 6 endroits où le temps a été observé sont reprérés par une flèche dans le programme %E3. Les quatre premières colonnes du tableau contiennent les informations habituelles obtenues lors d'un prélèvement du temps. (Cfr.: description des colonnes 1,2,3,4 en 3.2.2.2.). Quant à la colonne 5 ( $\Delta$ 1) elle contient le temps calculé d'extraction qui est aussi le temps d'exécution du programme. Ce temps est obtenu en soustrayant le contenu de la colonne 2 (compte tenu de la colonne 1) entre les lignes 1 et 4. On constate donc un temps moyen d'extraction de 49.54 ticks, soit 0.99 sec.

Cette première modification n'a donc apporté aucune amélioration au monitoring. L'extraction dure, en effet, deux fois plus longtemps que dans la version précédente. C'est probablement le prix qu'il faut payer pour les concaténations et les change-



ments type "numeric" → type "string" que l'on effectue tout au long de ce programme.

- 3.3.2. On a ensuite séparé l'extraction et la construction des deux strings nécessaires au stockage des données. Pour cela, on a repris la même manière de procéder pour extraire les informations du système que dans la première version du programme de surveillance (%LB). C'est cette nouvelle version de l'extraction qui est présentée dans le programme %E4. Nous avons, à nouveau, procédé à la mesure du temps d'exécution de ce programme. Les résultats de cette mesure sont présentés dans le tableau TAB - 3.3.2. Pour cette expérience, on a réduit à trois le nombre de prélèvements du temps au cours d'une exécution. Les colonnes  $\Delta 1$  et  $\Delta 2$  du TAB - 3.3.2 représentent respectivement le temps d'extraction des données et le temps de stockage sur disque de ces données.

Le temps moyen d'extraction est de 24.66 ticks, soit 0.493 sec. et le temps moyen de stockage est de 38.77 ticks, soit 0.775 sec. En soustrayant le temps utilisé pour observer le temps, soit 0.063 sec., on obtient la paire de résultats (0.430sec/0.712 sec.) qui est à comparer avec la paire de résultats (0.455sec/3.910 sec.) obtenu en 13 - 3 au paragraphe précédent.

- 3.3.3. On a ensuite repris le programme de surveillance %LB qui consommait beaucoup de CPU et on a essayé de le privilégier par rapport aux autres utilisateurs "CPU Bound". Pour cela, on a songé à modifier la valeur de la tranche temps courante au moment où on commençait l'extraction des données. Pour que cette modification soit effective, il fallait s'assurer que les entrées de SYSTAB :

- a) SYSTAB + 30 : nombre de ticks restant dans la tranche de temps courante.
- b) SYSTAB + 31 : nombre de ticks attribués à la tranche de temps courante.

étaient réellement les endroits qui étaient consultés par la "system queuing directive" TIMEUP (voir appendice MUMPS-11 system), activée par l'interpréteur et activant la procédure "SWAP OUT" (voir également MUMPS-11 system appendice) en cas d'épuisement de la tranche de temps.

On a pour cela repris le programme %E2 auquel on a apporté les modifications suivantes :

- a) en 1.30 on a supprimé la commande H  $\emptyset, \emptyset, \emptyset$  ce qui nous épargnait le temps de passage de "High wait queue" en "Low wait queue".
- b) en 1.25 on effectuait les commandes suivantes :

A 46 P 256 V T2:8115 U 46 S L=L+1 D 10

par lesquelles on fixe à 35 le nombre de ticks attribués à la tranche de temps courante et



le nombre de ticks restant dans la tranche de temps courante.

et on fait un prélèvement du temps.

L'exécution du programme %E2, ainsi modifié, fournissait les résultats présentés dans le tableau TAB - 3.3.3.

Dans ce tableau, une exécution du programme est décrite par un groupe de 4 lignes. Les quatre premières colonnes ont leur signification habituelle (cfr.: paragraphe 3.2.2.2.).

La colonne  $\Delta$  1 représente le temps d'extraction calculé à partir de S28. La colonne  $\Delta$  2 a la même signification que  $\Delta$  1, mais son contenu est calculé à partir de S30. Quant à la colonne  $\Delta$  3 elle représente le temps d'analyse. Les temps moyens obtenus pour le temps d'extraction et le temps d'analyse sont respectivement de 0.520 sec. et 3.931 sec., déduction faite du temps d'observation du temps. On peut à nouveau constater la cohérence de cette paire de résultats avec la paire obtenue au paragraphe 3.2.2.2. La différence, toutefois, d'environ 3 ticks pour le temps d'extraction suggère tout de même le fait que l'on ait pu gagner 1 à 2 ticks pour l'extraction en évitant l'overhead du système provoqué par l'extraction de la séquence SWAP OUT/IDLE LOOP/SWAP IN.

3.3.4. On a ensuite refait l'expérience précédente dans la condition où on n'était plus le seul utilisateur du système. Les résultats présentés dans le tableau TAB - 3.3.4 montre que si on est privilégié par rapport aux utilisateurs "CPU Bound", on reste malgré tout victime de toutes les interruptions "hardware" du système. Aussi, dans l'étape suivante, avons-nous tenté de dominer de manière software le système d'interruption du processeur PDP-11.

3.3.5. Le processeur PDP-11 possède 8 registres utilisés pour l'exécution de programme. Les 6 premiers, notés R0  $\rightarrow$  R5 sont utilisés à la guise du programmeur.

Les registres R6 et R7 sont conventionnellement utilisés comme "stack pointer" et "program counter". A côté de ces 8 registres, on trouve également un certain nombre d'autres registres dont le Processor Status Word qui va nous intéresser par la suite.

Au niveau MUMPS-11, la consultation de la documentation sur le système d'exploitation indique qu'au cours de la procédure SWAP OUT, sont sauvées dans une certaine zone de la partition (STACK USER) toutes les informations nécessaires au redémarrage du job lorsqu'à nouveau celui-ci aura le contrôle du



processeur.

Au niveau du PDP-11 processor [18] il est indiqué que lorsqu'une commande d'interruption est enregistrée par le processeur, une adresse est transmise au processeur. C'est l'adresse d'une zone de 2 mots mémoire qui contiennent le nouveau Program Counter et le nouveau Processor Status Word. Les PC et PSW courants sont alors sauvés dans des zones temporaires (en fait, le sommet du stack system).

A la fin du service de l'interruption, on retrouve les anciens PC et PSW qui redeviennent courants et qui permettent la reprise du travail interrompu. Toutes les interruptions pour le processeur PDP-11 ne sont pas équivalentes. Un système de priorité a été établi. Cette priorité se trouve dans les bits 5,6,7 du Processor Status Word.

Dans le cadre de l'installation PDP-11 d'Interactive System, on peut représenter le système prioritaire d'interruptions par le schéma suivant :

Niveau 6 : horloge.

Niveau 5 : disque et bande magnétique.

Niveau 4 : imprimante et vidéo.

Niveau 0 : programme utilisation.

Pour le programme de monitoring, il s'agirait de pouvoir masquer le Processor Status Word durant l'extraction afin d'éliminer tout risque d'interruptions sans que cela gêne la mise à jour du temps dans SYSTAB. On a pour cela écrit le programme %B2.

3.3.5.1. Pour la première utilisation du programme %B2, on a voulu voir quel serait l'effet résultant du fait que l'on place la priorité 6 dans le PSW au cours de l'exécution d'un programme. Les résultats de l'exécution de %B2 présentés en TAB - 3.3.5 montre clairement que tant que le PSW a la priorité 6, le temps qui s'écoule n'est plus mis à jour sans SYSTAB.

3.3.5.2. On a ensuite exécuté le programme %B2 en forçant la priorité 5 dans le PSW. Pour ce faire, il faut modifier le step 1.20 en

..... V 65534: \$V(65534) + 160 ....

Les résultats de cette exécution sont présentés en TAB - 3.3.6. On peut maintenant constater que le temps qui s'écoule est mis à jour dans SYSTAB, mais en plus, et c'est plus grave, on voit surtout que la priorité 5 n'est pas maintenue dans le PSW.

On voit ainsi que la seule possibilité d'obtenir une image unique du système, en masquant les interruptions du processeur, consiste à

à forcer la priorité 6 dans le PSW durant le temps d'extraction des informations relatives à l'état du système.

### 3.3.6. CONCLUSION.

Une amélioration réalisable du programme %LB utilisé pour effectuer un monitoring de l'installation MUMPS-11 de la Banque de Bruxelles consisterait donc en la substitution (%LB) → (%L2,%L3) où:

%L2 serait l'extracteur des informations relatives à l'état du système et dans lequel la commande H 0,0,0 de %LB serait supprimée au profit de A 46 P256 VT2:8995 U46.

%L3 serait l'analyseur des données extraites. Néanmoins, il faut se rendre compte que cette solution ne permet toujours pas d'obtenir à tout coup une image UNIQUE du système. Cette image unique du système ne peut être prélevée qu'en forçant la priorité 6 dans le Processor Stats Word, solution qui ne pourrait être adaptée<sup>que</sup> dans le cas où on aurait la certitude qu'aucun job n'utiliserait le temps fournit par la machine.



\$T	S28	S30	S31	$\Delta$ 1
51476	43	13	15	48
76	24	0	15	
76	4	0	15	
77	41	0	15	
77	30	2	4	
86	50	4	5	
86	43	13	15	48
86	24	0	15	
86	4	0	15	
87	41	0	15	
87	30	2	4	
96	50	4	5	
96	43	13	15	49
96	24	0	15	
96	5	0	15	
97	42	0	15	
97	32	1	4	
506	50	4	5	
06	43	13	15	49
06	24	0	15	
06	5	0	15	
07	42	0	15	
07	31	1	4	
16	50	4	5	
16	43	13	15	50
16	25	0	15	
16	6	0	15	
17	43	0	15	
17	33	1	4	
26	50	4	5	
26	43	14	15	50
26	25	0	15	
26	6	0	15	
27	43	0	15	
27	33	2	4	
36	50	4	5	
36	43	14	15	51
36	25	0	15	
36	6	0	15	
37	44	0	15	
37	33	2	4	
46	50	4	5	
46	44	14	15	50
46	25	0	15	
46	7	0	15	
47	44	0	15	
47	34	2	4	
56	50	4	5	

TAB - 3.3.1.

\$T	S28	S30	S31	$\Delta_1$	$\Delta_2$
39355 55 56	35 8 32	13 0 2	15 15 4	27	26
65 65 66	48 24 46	14 0 2	15 15 4	24	28
75 75 76	48 24 44	14 0 2	15 15 4	24	30
85 85 86	48 24 38	14 0 2	15 15 4	24	36
95 95 96	48 24 35	13 0 2	15 15 4	24	39
405 05 06	48 23 35	13 0 2	15 15 4	25	38
15 15 16	48 23 33	13 0 2	15 15 4	25	40
25 25 26	48 23 28	13 0 2	15 15 4	25	45
35 35 36	48 23 26	13 0 2	15 15 4	25	47

TAB - 3.3.2.



\$T	S28	S30	S31	$\Delta$ 1	$\Delta$ 2	$\Delta$ 3
97	46	33	35	29	29	199
97	17	4	35			
101	18	0	35			
08	49	4	5			
08	45	33	35	28	29	200
08	17	4	35			
12	17	0	35			
19	49	4	5			
19	45	33	35	29	29	199
19	16	4	35			
23	17	0	35			
30	49	3	5			
30	45	33	35	29	29	200
30	16	4	35			
34	16	0	35			
41	49	3	5			
41	45	33	35	29	29	200
41	16	4	35			
45	16	0	35			
52	49	3	5			
52	45	33	35	30	30	200
52	15	3	35			
56	15	0	35			
63	49	3	5			
63	45	33	35	30	30	200
63	15	3	35			
67	15	0	35			
74	49	3	5			

TAB - 3.3.3.

\$T	S28	S30	S31	$\Delta 1$	$\Delta 2$	$\Delta 3$
38661	7	34	35	33	?	180
662	24	15	5			
666	44	2	15			
671	49	4	5			
671	45	33	0	35	?	182
671	10	0	15			
675	28	1	15			
682	49	4	5			
682	45	33	35	29	30	183
682	16	3	35			
686	33	4	10			
692	49	4	5			
692	45	33	35	29	29	200
692	16	4	35			
696	16	11	15			
703	49	3	5			
703	45	33	35	30	30	170
703	15	3	35			
707	45	0	5			
713	49	3	5			
713	45	33	35	30	30	178
713	15	3	35			
717	37	2	10			
723	49	3	5			
723	45	33	35	30	30	153
723	15	3	35			
726	12	14	15			
734	49	3	5			
734	44	32	35	35	?	185
734	9	1	5			
738	24	13	15			
745	49	3	5			

TAB - 3.3.4.





\*\*\*\*\* OBSERVATION DU TEMPS. \*\*\*\*\*

70330	17	2	5
70330	14	10	10
70330	13	8	10
70330	11	6	10
70330	9	4	10

\*\*\*\*\* OBSERVATION DU PROCESSOR STATUS WORD. \*\*\*\*\*

PSW= 1            PRIORITE= 0

\*\*\*\*\* MODIFICATION DU PROCESSOR STATUS WORD. \*\*\*\*\*

\*\*\*\*\* OBSERVATION DU PROCESSOR STATUS WORD. \*\*\*\*\*

PSW= 161            PRIORITE= 5

\*\*\*\*\* OBSERVATION DU TEMPS. \*\*\*\*\*

70330	3	14	15
70330	1	12	15
70331	49	9	15
70331	47	7	15
70331	44	5	15
70331	42	2	15
70331	40	15	15
70331	37	12	15
70331	35	10	15
70331	32	7	15

\*\*\*\*\* OBSERVATION DU PROCESSOR STATUS WORD. \*\*\*\*\*

PSW= 1            PRIORITE= 0

\*\*\*\*\* MODIFICATION DU PROCESSOR STATUS WORD. \*\*\*\*\*

\*\*\*\*\* OBSERVATION DU PROCESSOR STATUS WORD. \*\*\*\*\*

PSW= 1            PRIORITE= 0

\*\*\*\*\* OBSERVATION DU TEMPS. \*\*\*\*\*

70331	26	2	15
70331	24	14	15
70331	21	11	15
70331	18	9	15
70331	16	6	15



### 3.4. SURVEILLANCE D'UNE MEME CHARGE PAR DEUX VERSIONS DU PROGRAMME.

Si, selon la conclusion du paragraphe précédent, il semble difficile d'améliorer le programme de surveillance de manière à lui permettre de prélever une image unique du système, il n'est cependant pas illusoire de vouloir atteindre l'image unique du système. Pour illustrer cette préoccupation, on a réalisé l'expérience suivante :

- a) on charge 7 partitions du système avec le même programme : %B3.
- b) on utilise la huitième partition pour effectuer la surveillance de cette charge : (%L2,%L3).

Cette expérience s'est ainsi déroulée dans deux conditions de surveillance :

- a) un programme extracteur (%L2) ne permettant pas l'obtention d'une image unique du système. Ce type de surveillance sera référencé, par la suite, MONITORING-1.
- b) une version modifiée de %L2 (%MB) permettant l'extraction d'une image unique du système. Cette surveillance sera, quant à elle, référencée MONITORING-2.

On présente dans TAB - 3.4.1, un récapitulatif du MONITORING-1 et dans TAB - 3.4.2. celui de MONITORING-2. On peut constater tout d'abord que l'état 32. a complètement disparu dans MONITORING-2. C'est une confirmation de l'hypothèse émise au chapitre II.

On peut ensuite remarquer que bien qu'elles soient chargées identiquement, les partitions présentent chacune un aspect différent quant au résultat de leur différence. Il est probable que cette différence doit s'atténuer au fur et à mesure que la taille de l'échantillonnage augmente. On peut également remarquer que la "Low Wait Queue" n'est pas occupée dans les deux cas. Ce résultat est normal étant donné que l'utilisation de la ressource CPU ne requiert pas le passage dans la Low Wait Queue. Quant aux taux d'occupation des différents états, ils sont tout-à-fait inattendus lorsqu'on compare les chiffres obtenus pour une même partition surveillée dans les conditions MONITORING-1 et MONITORING-2.

Les diagrammes rassemblés en TAB - 3.4.3., TAB - 3.4.4. et TAB - 3.4.5. présentent la distribution du nombre d'occurrences de chaque état observé dans chaque ligne de monitoring.

On peut remarquer une chose importante dans les conditions de MONITORING-2 : la distribution de la population des files d'attente 42 (SHORT QUEUE) et 43 (DISK I/O BOUND QUEUE) apparaît maintenant cohérente avec la définition de chacune de ces deux files d'attente.

C'est aussi une confirmation de l'hypothèse émise au chapitre II.

En conclusion, cette expérience a fait apparaître clairement les déficiences de la surveillance lorsqu'on la

réalisait dans les conditions MONITORING-1.

Les résultats de cette expérience sont aussi significatifs dans la mesure où, lorsqu'on passe d'une condition de surveillance à une autre, les résultats peuvent passer du vrai au faux (file d'attente 47) et où, s'il y a seulement une erreur en passant d'une surveillance à une autre, la mesure de cette erreur est difficilement estimable.



ETAT	1	2	3	4	5	6	7	8		
28	/	/	/	/	/	/	/	/	/	/
29	/	/	/	/	/	/	/	/	/	/
30	/	/	/	/	/	/	/	/	/	/
32	18.21	20.15	17.05	23.64	17.44	/	15.89	15.89	16.03	1.28
41	/	/	/	/	/	100	/	/	12.50	1.00
42	1.55	1.16	0.77	1.16	1.55	/	2.32	1.55	1.25	0.10
43	16.66	22.86	24.41	26.74	29.06	/	27.51	26.35	21.70	1.73
44	10.46	11.62	8.52	6.97	8.91	/	7.36	7.75	7.70	0.61
45	/	/	/	/	/	/	/	/	/	/
46	/	/	/	/	/	/	/	/	/	/
47	/	/	/	/	/	/	/	/	/	/
48	1.93	1.55	1.55	1.16	1.93	/	1.55	4.26	1.74	0.13
49	/	/	/	/	/	/	/	/	/	/
50	51.16	42.63	47.67	40.31	41.08	/	45.34	44.18	39.05	3.12
51	/	/	/	/	/	/	/	/	/	/

TAB - 3.4.1.

ETAT	1	2	3	4	5	6	7	8		
28	/	/	/	/	/	/	/	/	/	/
29	/	/	/	/	/	/	/	/	/	/
30	/	/	/	/	/	/	/	/	/	/
32	/	/	/	/	/	/	/	/	/	/
41	/	/	/	/	/	/	100.00	/	12.50	1.00
42	1.01	0.50	2.02	0.50	/	1.51	/	2.52	1.01	0.08
43	8.08	9.59	4.04	7.57	6.06	7.57	/	8.58	6.43	0.51
44	1.01	3.03	1.01	1.51	2.52	0.50	/	2.02	1.45	0.11
45	/	/	/	/	/	/	/	/	/	/
46	/	/	/	/	/	/	/	/	/	/
47	59.59	52.02	60.60	64.64	54.54	58.58	/	58.08	51.01	4.08
48	1.51	4.54	4.04	2.02	4.04	1.51	/	4.54	2.77	0.22
49	/	/	/	/	/	/	/	/	/	/
50	28.78	30.30	28.28	23.73	32.82	30.30	/	24.24	24.81	1.98
51	/	/	/	/	/	/	/	/	/	/

TAB - 3.4.2.



## TAB-3.4.3

FILE D'ATTENTE 32 [EN TRANSITION]

0	9	**
1	168	*****
2	80	*****
3	1	*

— MONITORING - 1 —

FILE D'ATTENTE 32 [EN TRANSITION]

0	198	*****
---	-----	-------

— MONITORING - 2 —

FILE D'ATTENTE 50 [CLOCK QUEUE]

0	1	*
1	3	*
2	44	*****
3	139	*****
4	58	*****
5	12	***
6	1	*

— MONITORING - 1 —

FILE D'ATTENTE 50 [CLOCK QUEUE]

0	2	*
1	15	*****
2	165	*****
3	16	*****

— MONITORING - 2 —

## TAB-3.4.5

## FILE D'ATTENTE 42 [SHORT QUEUE]

0	243	*****
1	9	**
2	1	*
3	5	*

## FILE D'ATTENTE 43 [DISK I/O BOUND QUEUE]

0	20	*****
1	93	*****
2	88	*****
3	50	*****
4	6	**
5	1	*

## FILE D'ATTENTE 44 [DISK RESOURCE BOUND QUEUE]

0	179	*****
1	17	****
2	47	*****
3	14	****
6	1	*

— MONITORING - 1 —

## FILE D'ATTENTE 42 [SHORT QUEUE]

0	182	*****
1	16	*****

## FILE D'ATTENTE 43 [DISK I/O BOUND QUEUE]

0	96	*****
1	102	*****

## FILE D'ATTENTE 44 [DISK RESOURCE BOUND QUEUE]

0	182	*****
1	13	****
2	2	*
6	1	*

— MONITORING - 2 —



### 3.5. ETUDE DES VARIATIONS DU TEMPS D'EXECUTION.

Nous avons finalement essayé de préciser comment variait la durée d'exécution du programme de monitoring en fonction de la charge du système et du type de charge du système.

Nous avons, pour cela, envisagé trois types de charge :

- a) un programme simulant une séquence de tâches interactives → ce qui va constituer une charge réaliste du système (%B3).
- b) un programme ne requérant que la ressource CPU → ce qui va constituer une charge non réaliste du système (%L8).
- c) un programme ne réalisant que des I/O disques ce qui, également, constitue une charge non réaliste du système (%L9).

3.5.1. On a tout d'abord effectué des mesures du temps d'exécution sur le programme %LB (version initiale du programme de surveillance). On a modifié ce programme en ajoutant :

- 1) une procédure d'observation du temps.
- 2) une procédure contrôlant la charge du système et effectuant pour chaque charge, dix mesures du temps d'exécution (temps d'extraction et temps d'analyse).

Cette version modifiée du programme de surveillance est référencée %L7.

Les résultats concernant une charge du système constituée par le programme %B3 sont rassemblés dans le tableau TAB - 3.5.1. La signification des différentes colonnes de ce tableau est la suivante :

- colonne 1 : charge du système exprimée en nombre de partitions actives.
- colonne 2 : temps d'extraction exprimé en ticks
- colonne 3 : temps d'extraction exprimé en sec.
- colonne 4 : temps d'analyse exprimé en ticks.
- colonne 5 : temps d'analyse exprimé en sec.
- colonne 6 : temps total d'exécution en ticks, calculé en faisant la somme des colonnes 2 et 4.
- colonne 7 : temps total exprimé en secondes.

TAB - 3.5.2. est relatif aux mesures effectuées avec une charge constituée par le programme %L8 et TAB - 3.5.3 pour le programme %L9.

Ces résultats sont illustrés par les graphiques référencés fig - 3.5.1., fig - 3.5.2. et fig - 3.5.3. On peut observer la même tendance sur chacune des figures. L'absence de variation du temps d'exécution, lorsque la charge du système est essentiellement orientée accès disque, est surprenante. L'explication que l'on peut donner à ces mesures est la suivante : on peut considérer que le système



voit sa charge constituée d'une part d'un programme de surveillance essentiellement "CPU Bound" et d'autre part d'un programme de la charge qui, lui, est essentiellement "I/O disk bound".

En effet, le système MUMPS-11 est dessiné pour privilégier le travail qui effectue une E/S disque et pour ce faire, on a créé la paire de files d'attente (SHORT QUEUE/DISK I/O BOUND QUEUE) dont la politique d'occupation est la suivante :

- a) le job pour lequel une E/S disque est en train d'être réalisée, est placé en DISK I/O BOUND QUEUE.
- b) Lorsque cette E/S disque est terminée, le job est placé en SHORT QUEUE (qui est la file d'attente de plus haute priorité) de manière à ce que le job puisse continuer l'exécution des instructions qui suivent l'E/S disque.
- c) Lorsque le job est placé en SHORT QUEUE, le système laisse la DISK I/O BOUND QUEUE dans un état non utilisable pour le cas où une instruction d'E/S suivrait celle qui vient d'être terminée.

Il est dès lors clair qu'il ne peut jamais y avoir plus de 1 job dans chacune de ces files d'attente. Aussi, pour la charge envisagée, le processeur partage son contrôle entre le programme de surveillance et le programme de la charge qui se trouve en SHORT QUEUE. Le processeur "voit" toujours la même chose qu'il y ait 1 ou 7 partitions occupées.

Ce qui n'apparaît toutefois pas explicable, c'est le fait que le temps d'analyse (et dès lors le temps total d'exécution) diminue lorsque la charge du système augmente.

Quant à la très forte augmentation du temps d'analyse dans le cas de la charge "CPU ONLY", elle est compréhensible par le fait que le programme de surveillance et les programmes de la charge sont des concurrents de même poids pour l'obtention du contrôle du processeur. Dans ces conditions, en effet, seules les 2 "WAIT QUEUE" sont occupées par tous les jobs dans le système.

- 3.5.2. On a ensuite effectué la même série de mesures avec le programme %L2 auquel on a apporté les modifications décrites en 3.5.1.

Cette version modifiée de l'extracteur s'appelle %L6. Les résultats rassemblés dans TAB - 3.5.4., TAB - 3.5.5. et TAB - 3.5.6. sont présentés dans le même ordre que ceux de la mesure précédente.

Ces résultats sont illustrés par les graphiques référencés fig - 3.5.4., fig - 3.5.5. et fig - 3.5.6. La fig - 3.5.4. présente les mêmes variations que fig - 3.5.1., ce qui est normal puisque l'extrac-tion se fait de la même manière dans %L6 et %L7.



Pour fig - 3.5.5., on peut expliquer les mesures obtenues pour le stockage (qui nécessite deux I/O disque) dans un système chargé "DISK ACCESS ONLY" de la même manière que pour le temps d'analyse dans un système chargé "CPU ONLY". En effet, arrivé dans l'exécution de la procédure de stockage, le programme est mis sur pied d'égalité vis-à-vis de tous les autres programmes de la charge et doit transiter par les files d'attente "DISK RESOURCE BOUND QUEUE" (très occupée dans une charge "DISK ACCESS ONLY") "DISK I/O BOUND QUEUE" "SHORT QUEUE" pour terminer son exécution.

Par contre, %L6, plongé dans une charge "CPU ONLY" apparaît, avec ses deux I/O DISK, comme une tâche interactive parmi des tâches "CPU BOUND".

1	2	3	4	5	6	7
1	29.22	0.584	199.66	3.993	228.88	4.577
2	35.00	0.700	219.33	4.386	254.33	5.086
3	37.88	0.757	234.88	4.697	272.76	5.455
4	37.88	0.757	255.22	5.104	292.10	5.862
5	44.44	0.888	279.77	5.595	324.21	6.484
6	47.55	0.951	293.33	5.866	340.88	6.817
7	53.00	1.060	325.00	6.500	378.00	7.560

TAB - 3.5.1.

1	2	3	4	5	6	7
1	29.55	0.591	200.00	4.000	229.55	5.591
2	31.88	0.637	202.77	4.055	234.65	4.693
3	31.44	0.628	192.00	3.840	223.44	4.468
4	31.66	0.633	181.55	3.631	213.21	4.264
5	31.77	0.635	173.55	3.471	205.32	4.106
6	31.55	0.631	157.33	3.146	188.88	3.777
7	31.66	0.633	145.77	2.915	177.43	3.548

TAB - 3.5.2.

1	2	3	4	5	6	7
1	32.22	0.644	208.77	4.175	240.99	4.819
2	57.77	1.155	395.00	7.900	452.77	9.055
3	41.44	0.828	559.66	11.193	601.10	12.022
4	86.88	1.737	638.33	12.766	725.21	14.504
5	113.55	2.271	894.11	17.882	1007.66	20.153

TAB - 3.5.3.

1	2	3	4	5	6	7
1	24.44	0.488	46.55	0.931	70.99	1.419
2	25.22	0.504	49.44	0.988	74.66	1.493
3	30.11	0.602	51.44	1.028	81.55	1.631
4	32.66	0.653	62.66	1.253	95.32	1.906
5	30.55	0.611	70.88	1.417	101.43	2.028
6	36.83	0.736	70.00	1.400	106.83	2.136
7	38.33	0.766	82.88	1.657	121.21	2.424

TAB - 3.5.4.

1	2	3	4	5	6	7
1	23.66	0.473	32.88	0.657	56.14	1.122
2	25.33	0.506	54.55	1.091	79.88	1.597
3	25.44	0.508	482.88	9.657	508.32	10.166
4	25.55	0.511	712.33	14.246	737.88	14.757
5	25.00	0.500	968.11	19.362	993.11	19.862
6	26.00	0.520	1216.88	24.337	1242.88	24.857
7	25.44	0.508	1487.77	29.755	1513.21	30.264

TAB - 3.5.5.

1	2	3	4	5	6	7
1	20.33	0.406	46.44	0.928	66.77	1.335
2	35.11	0.702	74.66	1.493	109.77	2.195
3	50.00	1.000	100.00	2.000	150.00	3.000
4	65.11	1.302	136.44	2.728	201.55	4.031
5	80.00	1.600	153.22	3.064	233.22	4.664
6	96.11	1.922	178.33	3.566	274.44	5.488
7	110.66	2.213	244.33	4.886	354.99	7.099

TAB - 3.5.6.



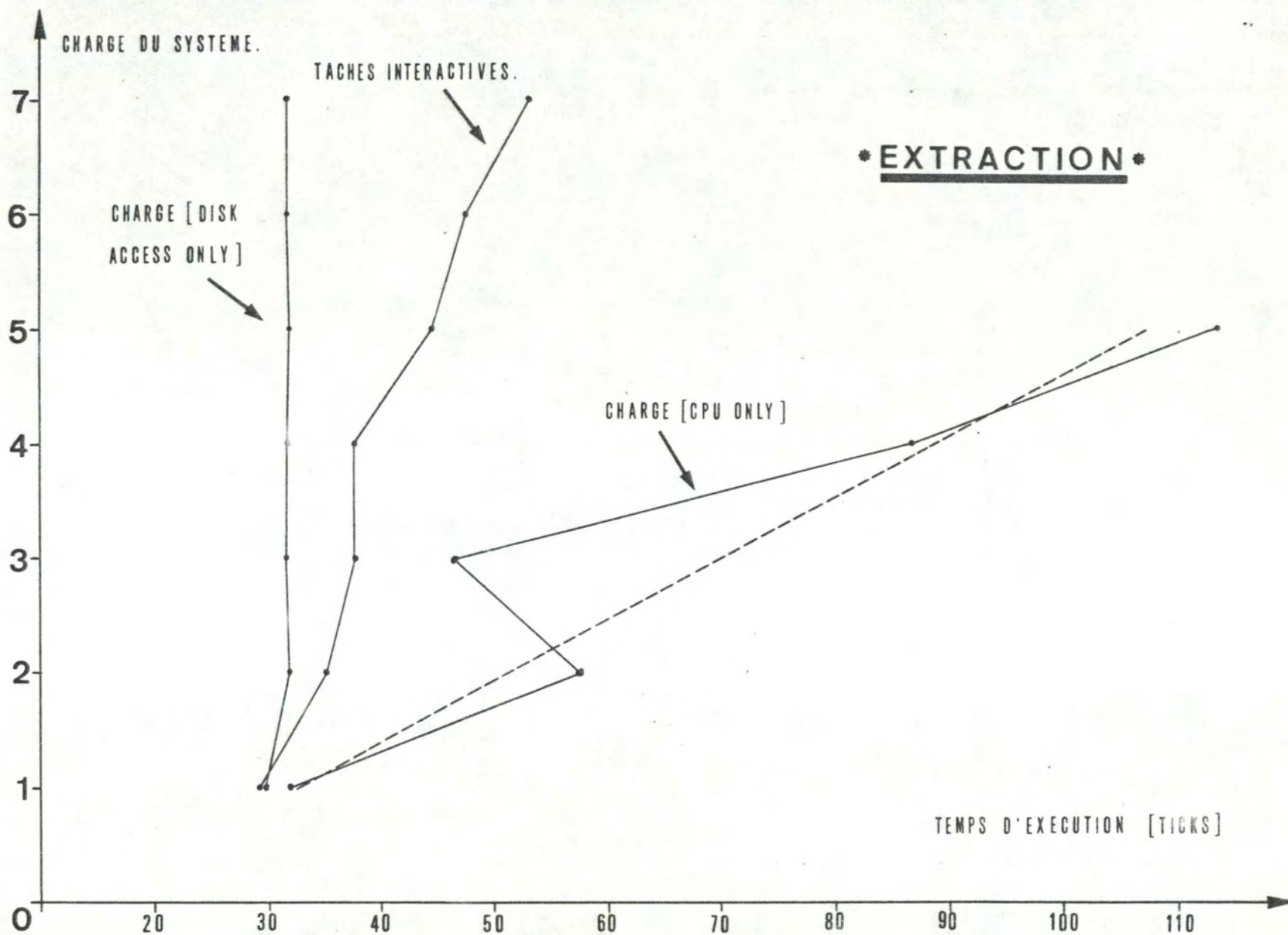


FIG-3.5.1

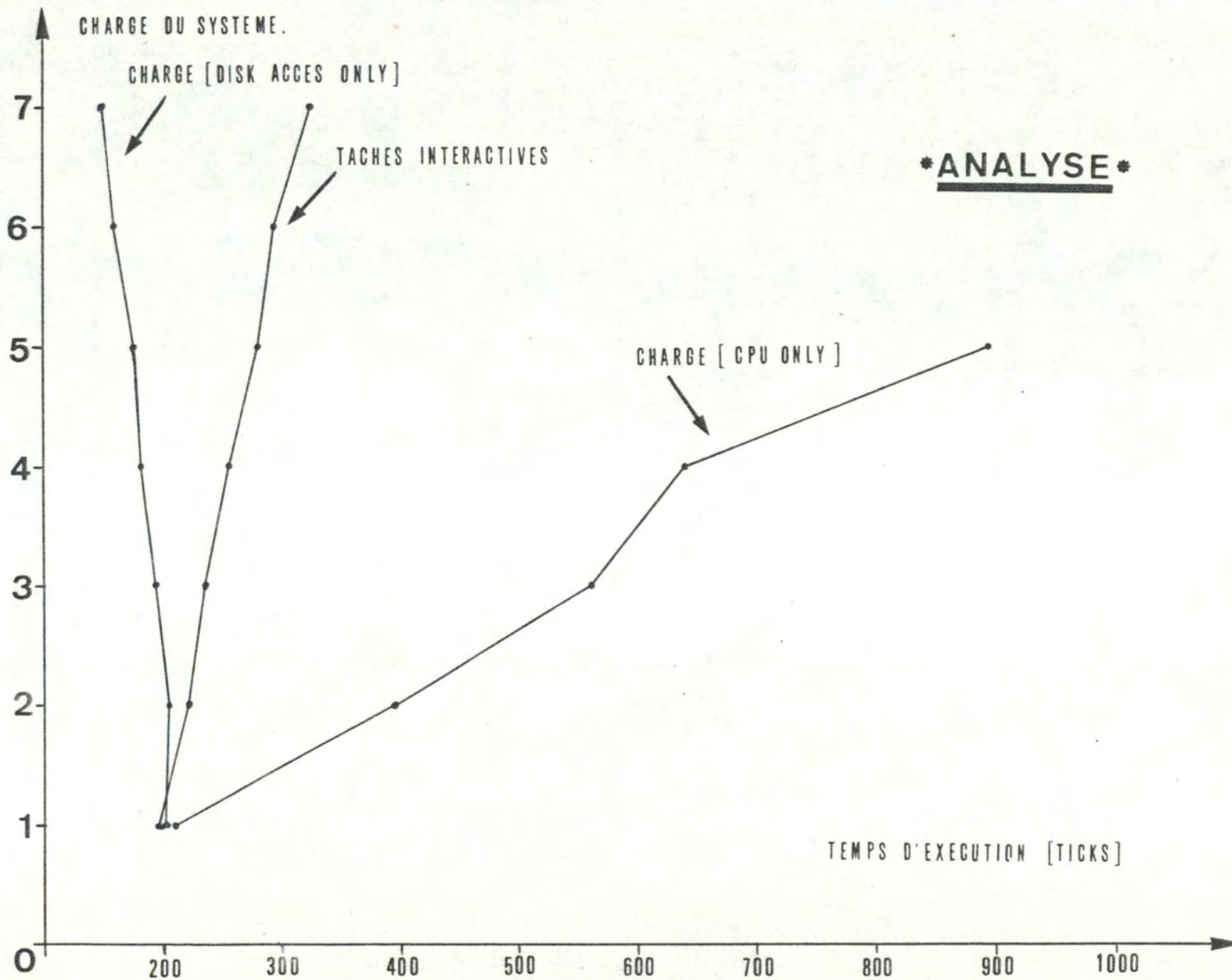


FIG-3.5.2



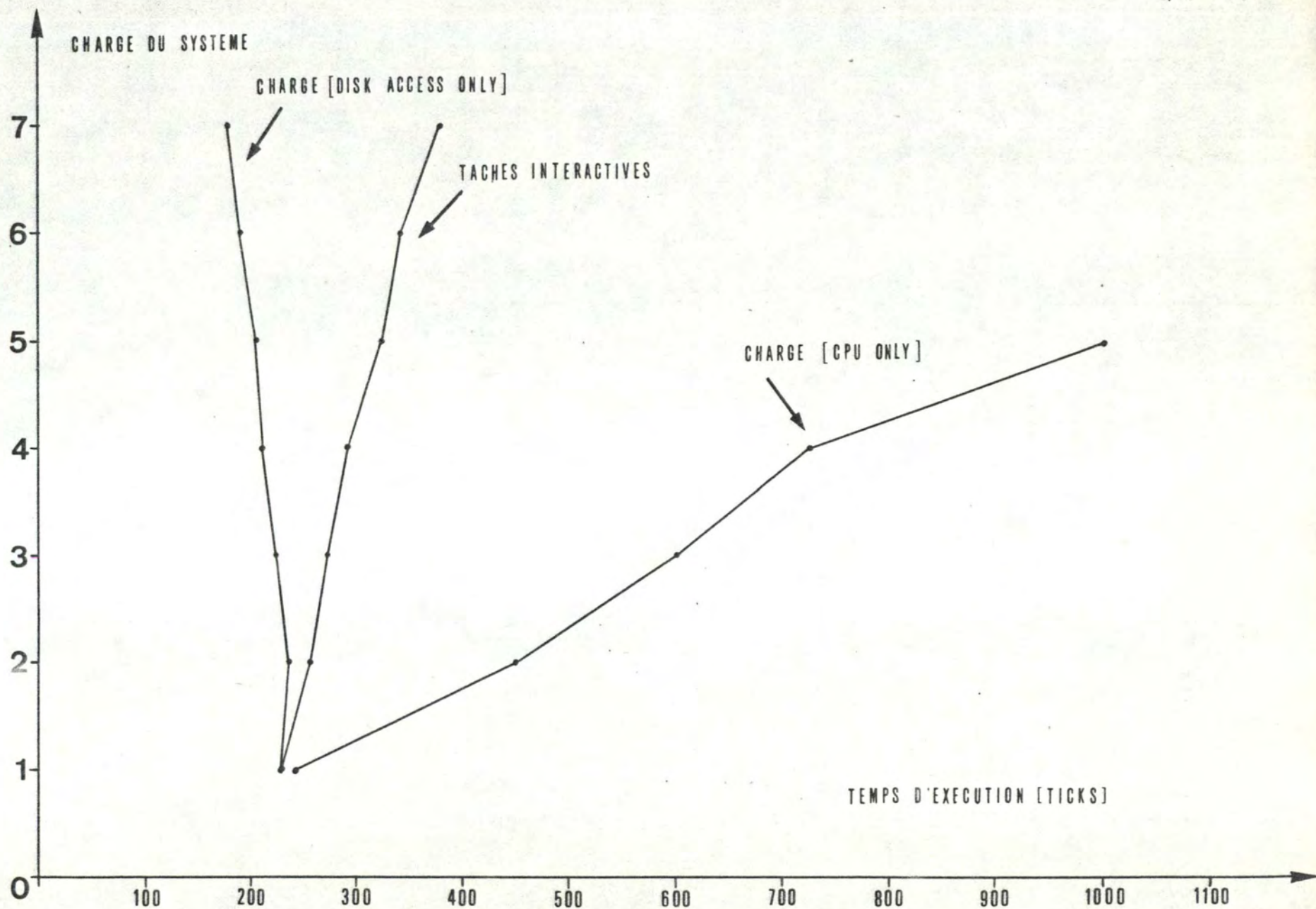


FIG-3.5.3

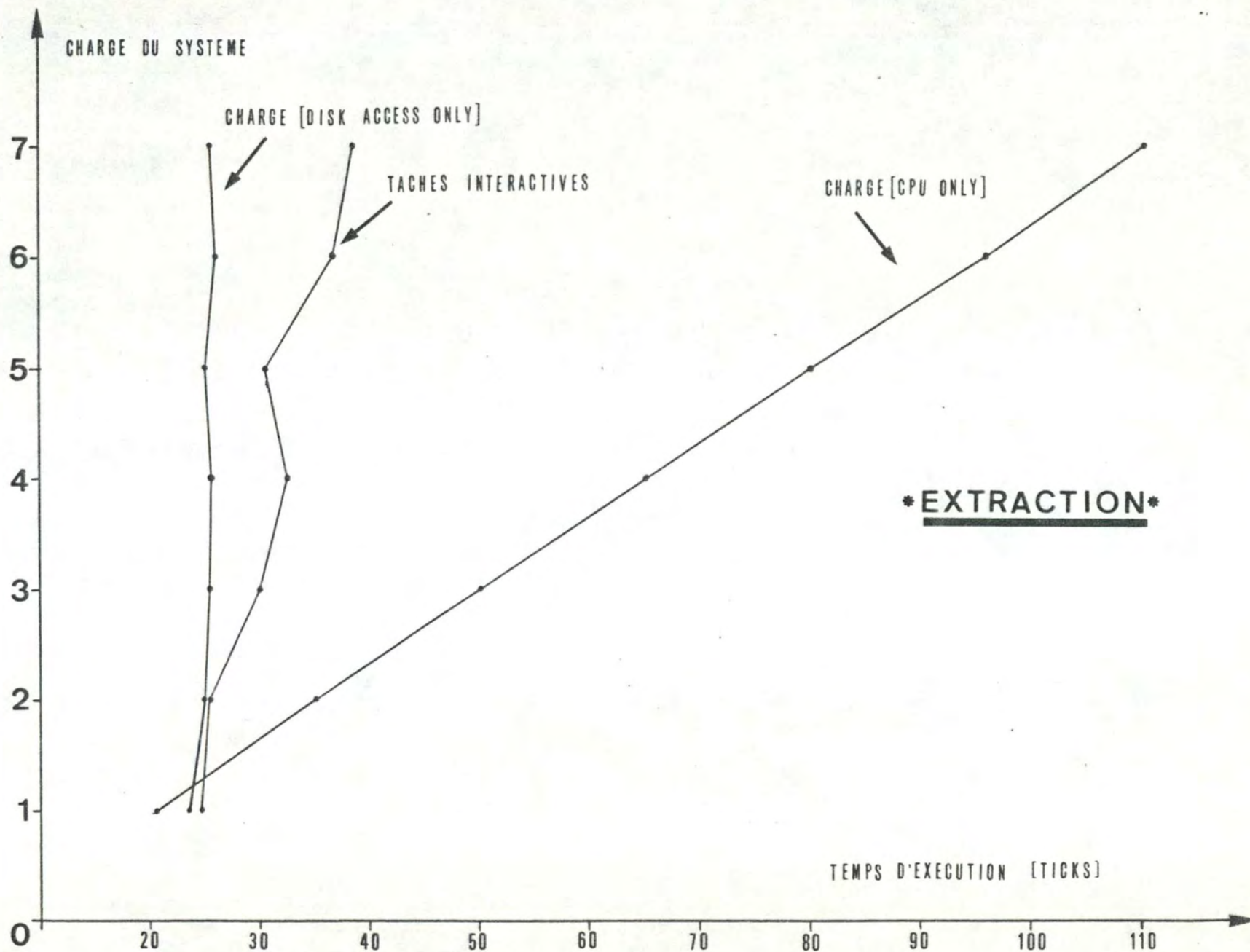


FIG-3.5.4



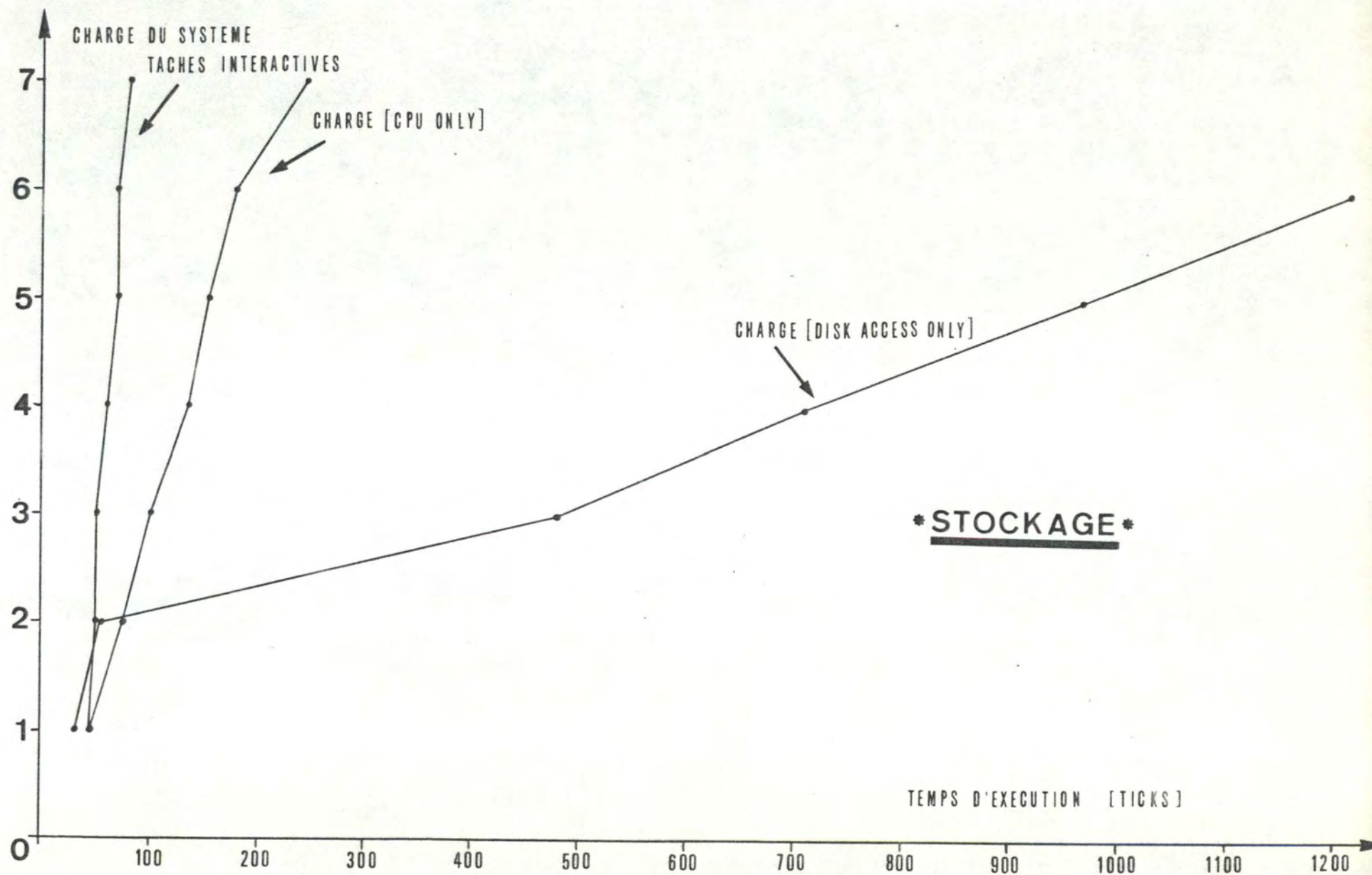


FIG-3.5.5

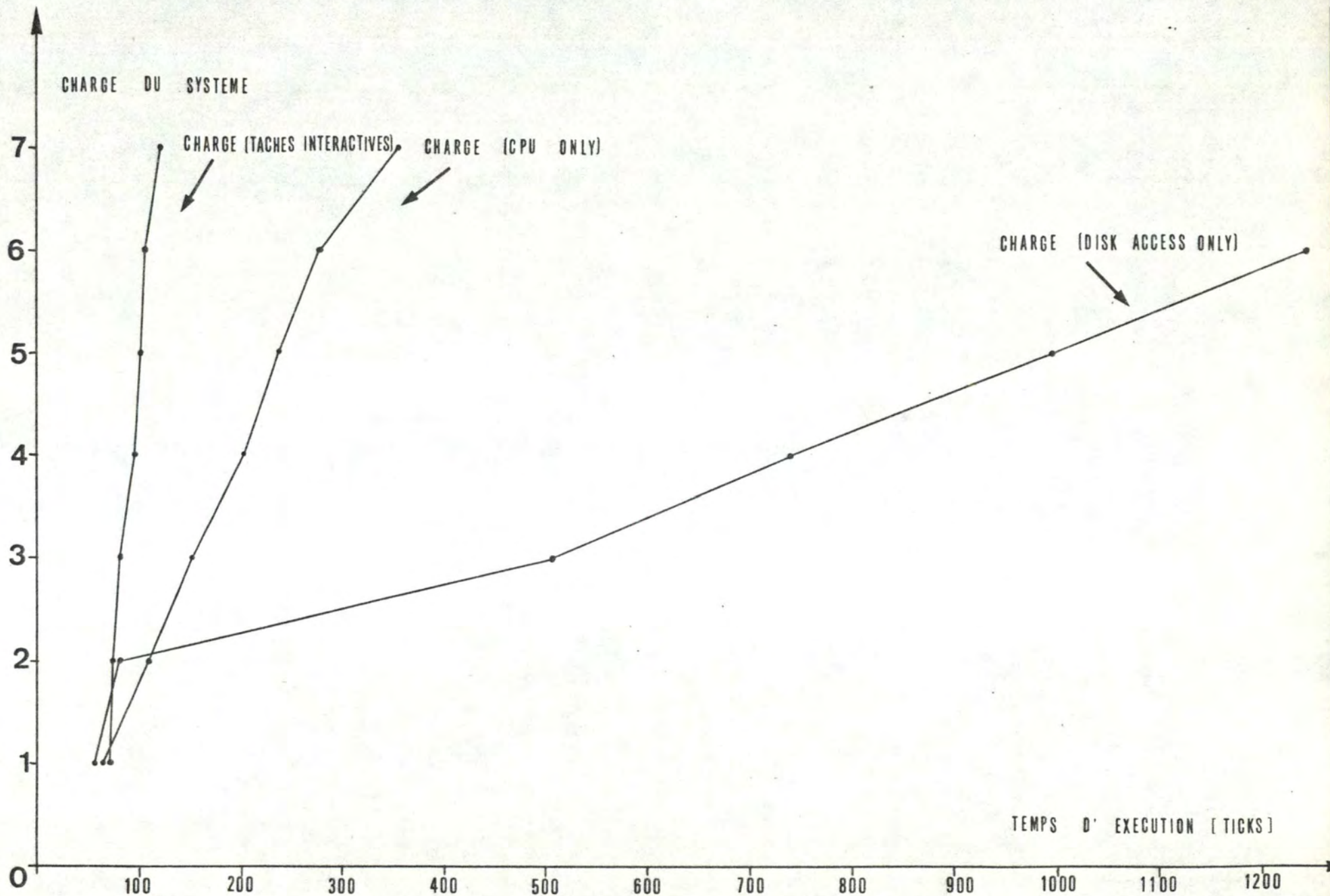


FIG-3.5.6



### 3.6. RESULTATS PARALLELES.

- 3.6.1. Dans cette première partie, on va décrire les informations issues des expériences présentées dans ce chapitre. Ce complément concerne le système des files d'attente. Pour chaque expérience de ce chapitre où il y a un prélèvement de temps, on a inclus les entrées
- SYSTAB + 30 : nombre de ticks restant dans la tranche de temps courante.
- SYSTAB + 31 : nombre de ticks attribués à la tranche de temps courante..
- Cela, dans le but d'approfondir la compréhension du système des files d'attente.

3.6.1.1. Le premier point que l'on a eu à éclaircir se situe dans les tableaux TAB-3.2.2. et TAB-3.2.3. (colonne S30) où on constate que, seul dans la machine et une fois une grande tranche de temps épuisée, SYSTAB + 30 n'est plus remis à jour pour une nouvelle décrémentation.

La réponse à cette question est la suivante :

- a) l'interpréteur génère une interruption pour déclencher la procédure "TIMEUP" (voir appendice C).
- b) la routine "TIMEUP" teste tout d'abord si l'utilisateur dispose encore de temps. S'il n'y a plus de temps, avant de faire un "SWAP OUT" de l'utilisateur, la procédure consulte l'entrée 154 de SYSTAB qui indique le nombre d'utilisateurs en attente du processeur. S'il n'y a aucun client en attente, le contrôle est rendu à l'interpréteur.
- c) Etant donné qu'ainsi on évite la séquence "SWAP OUT" / "IDLE LOOP" / "SWAP IN" et que c'est au cours de la procédure "SWAP IN" que sont initialisées SYSTAB + 30 et SYSTAB + 31, il est donc normal que SYSTAB + 30 reste à zéro une fois la première grande tranche de temps écoulée.

3.6.1.2. Le second point que l'on a observé concerne l'entrée SYSTAB + 31 (colonne S31) du tableau TAB-3.3.4. Il semble qu'un job qui a perdu le contrôle à la suite de la prise en charge d'une interruption par le processeur, et qui disposait d'une grande tranche de temps au moment de l'interruption, repasse par la séquence "HIGH WAIT QUEUE" / "MEDIUM WAIT QUEUE" / "LOW WAIT QUEUE" lorsqu'il reçoit à nouveau le contrôle du processeur. On peut comprendre cette politique si on veut privilégier le job interrompu vis-à-vis de tous les autres. D'autre part, si on se replace au niveau du programme de surveillance, cette politique détériore d'autant plus l'unicité du système.



3.6.1.3. Le troisième point observé concerne la valeur de la tranche de temps attribuée au job qui provient de la "SHORT QUEUE".

Dans la documentation relative au système MUMPS - 11, il est indiqué que cette valeur est de 1 tick. En fait, on a constaté que cette valeur est de 4 ticks.

3.6.2. Dans cette seconde partie, on tentera de quantifier les différentes façons d'envisager le prélèvement du temps.

3.6.2.1. Un job (%L5) a effectué l'observation du temps qui s'écoulait par rapport à lui.  
Les prélèvements étaient stockés dans l'espace de la partition qui était assigné à ce job.  
On a obtenu, en moyenne, 20 prélèvements par seconde.

3.6.2.2. Un job (%L4) a réalisé le même travail en sortant chaque prélèvement sur imprimante. On a obtenu, en moyenne cette fois, 2 prélèvements par seconde.

3.6.2.3. Un job (%B1) a effectué à nouveau le même travail en envoyant chaque prélèvement sur disque. On a ainsi obtenu en moyenne 4 observations par seconde.

3.6.2.4. C'est évidemment l'observation du temps avec stockage dans l'espace de la partition, qui est le procédé le plus riche. Ce procédé impose toutefois une limite (taille de la partition) très stricte aux prélèvements du temps.



## C H A P I T R E   I V   :

---

### EVALUATION DU SERVICE RENDU PAR .....

#### LE SYSTEME MUMPS - 11 .....

4.1. Introduction.

4.2. Mesure du temps de réponse.

4.3. Rôle de la méthode d'accès  
à la "data base" MUMPS - 11.

#### 4.1. Introduction.

Dans cette seconde partie du travail, on va plus spécialement s'attacher à l'évaluation du service rendu par un système TIME-SHARING [49].

Tout d'abord, dans cette introduction, on a essayé de déterminer un ensemble non limitatif de caractéristiques d'un système Time-Sharing.

Cet ensemble de caractéristiques peut être agencé selon la structure hiérarchique suivante :

##### 4.1.1. Accessibilité :

Pour cette catégorie, on désire mettre en évidence ce qui qualifie et quantifie l'accès à une installation orientée Time-Sharing. On distingue à cet effet:

##### 4.1.1.1. La disponibilité que l'on peut caractériser par:

- le temps de fonctionnement de l'installation ("useful up-time").
- La probabilité d'un succès lorsqu'on désire entrer en communication avec la machine pendant le temps de fonctionnement de l'installation.

##### 4.1.1.2. Les moyens dont on dispose pour approcher la machine. Ces moyens peuvent être caractérisés comme suit :

- le nombre de types de terminaux acceptés par la machine.
- La disponibilité de ces terminaux que l'on peut mesurer par le temps d'attente pour les utiliser.

##### 4.1.2. Les facilités d'emploi :

Pour cette catégorie, on veut faire apparaître les éléments qui sont susceptibles d'augmenter la qualité de l'installation et du service rendu.

##### 4.1.2.1. D'une manière générale, la qualité du service rendu sera fonction de l'existence de :

- facilités d'édition.
- facilités de travail en mode "BATCH".
- "debugging aids".
- facilités de manipulation de la "data base".

##### 4.1.2.2. La qualité du service dépend également du temps de réponse associé à différentes requêtes d'utilisateurs.

##### 4.1.2.3. La qualité de l'installation peut être mise en évidence par la répartition des types d'utilisateurs dont l'installation peut s'accomoder et la variété d'utilisations offerte par l'installation.

##### 4.1.2.4. Cette caractéristique très subjective que l'on pourrait intituler "Human Interface" met également en évidence la qualité d'une installation. Il s'agit du temps nécessaire à un utilisateur pour comprendre



le système et pour l'interprétation des divers incidents pouvant survenir au cours d'une journée de travail.

#### 4.1.3. La gestion de l'installation :

La qualité offerte pour la gestion de l'installation englobe les points suivants :

- 4.1.3.1. La détermination du coût d'utilisation des "facilities" offerte par le système.
- 4.1.3.2. Les possibilités de contrôle, pour le "manager" de l'installation, des accès aux divers mécanismes et ressources de l'installation.  
La possibilité de comptabiliser ces diverses utilisations.
- 4.1.3.3. L'extensibilité de l'installation qui peut être caractérisé par la mesure de la capacité courante, la facilité de modifier le software et la possibilité d'exécuter ce qui existe sur un hardware modifié.
- 4.1.3.4. La maintenance qui sera qualifiée par une mesure de la fiabilité de l'installation (nombre moyen de pannes par unité de temps) et par la mesure du temps moyen nécessaire à la réparation des pannes.

Ces trois catégories générales fournissent un moyen raisonnable (pas nécessairement le seul) de grouper les caractéristiques d'un système Time-Sharing.

La mesure de ces critères d'estimation dépend de la manière selon laquelle est utilisé le système Time-Sharing.

Le paragraphe 2 du présent chapitre sera consacré à la mesure du temps de réponse d'un accès disque pour diverses charges du système et pour diverses valeurs de la tranche de temps et le paragraphe 3 traitera de la manipulation de la data base MUMPS-11.



#### 4.2. Mesure du temps de réponse.

Ce paragraphe est consacré à la relation des résultats des mesures du temps de réponse effectuées dans le contexte MUMPS-11. La ressource disque étant la plus importante pour le système MUMPS-11, on a choisi la mesure du temps de réponse à un accès disque. A cet effet, on a construit une variable globale  $\uparrow$  REP à 4 niveaux (1 élément par niveau/l'élément à accéder). L'accès à la donnée du niveau 1 sera nommé par la suite type d'accès 1, de même on aura, pour le niveau 2, le type d'accès 2, pour le niveau 3, le type d'accès 3 et pour le niveau 4, le type d'accès 4.

Cette mesure de temps de réponse a été effectuée pour chaque charge du système (nombre de partitions actives) pour différents types de charges et pour différentes valeurs de la tranche de temps standard. Le programme %B4 a été écrit à cet effet. Ce programme effectue un certain nombre de fois la même mesure de temps de réponse pour différents types d'accès et différentes valeurs de la charge du système. Le programme %B4 contrôle la charge du système en effectuant, le moment voulu, le "START" du programme qui va constituer la charge du système.

4.2.1. Les deux premiers essais de mesures ont été peu significatifs. En effet, la charge utilisée pour ces mesures avait été construite sur le principe d'une utilisation des ressources disque et CPU.

4.2.1.1. La première charge utilisée fut la suivante :

1.20 F J=1:1:10 S DCA(J)= $\uparrow$ ACD(1,1,J)

1.30 F J=1:1:50 D 2

1.40 H 2 6 1.2

2.10 S T= $\$T$ , A= $\$V(\$V(44)+28)/100 \& 2.55 * 100$

2.20 S B= $\$V(\$V(44)+30)/100 \& 2.55 * 100$ , C= $\$V(\$V(44)+30)/25600 * 100$

Les résultats des mesures (qui ne se sont pas poursuivies au delà de la charge 4), effectuées seulement pour le type d'accès 1, sont les suivants :

X = charge du système (nombre de partitions actives)  
Y = temps de réponse en (sec/100)

X	1	2	3	4
Y	18.60	42.70	89.42	235.92

4.2.1.2. La seconde charge utilisée fut la suivante :

1.20 F J=1:1:10 S DCA(J)= $\uparrow$ ACD(1,1,J) D 2

1.30 H 2 6 1.2

2.10 S T= $\$T$ , A= $\$V(\$V(44)+28)/100 \& 2.55 * 100$

2.20 S B= $\$V(\$V(44)+30)/100 \& 2.55 * 100$ , C= $\$V(\$V(44)+30)/25600 * 100$

Les résultats obtenus pour le type d'accès 1 sont les suivants :



X	1	2	3	4	5	6	7	8
Y	18.30	51.60	248.10	495.80	721.96	973.26	1205.40	1447.36

4.2.2. Les résultats du paragraphe précédent ne correspondant à rien de ce qui avait pu être observé pour MUMPS-11, on a alors décidé de modifier le principe de construction de la charge. Etant donné que MUMPS-11 a été conçu pour un mode de travail interactif, on a cherché à simuler l'essentiel d'une tâche interactive, à savoir la séquence QUESTION/ATTENTE-REPONSE/TRAITEMENT DE LA REPONSE.

La "question" était représentée par un accès disque l'"Attente - Réponse" par une commande HANG, et le "Traitement de la réponse" par un traitement essentiellement "CPU BOUND".

On a ainsi écrit trois programmes destinés à réaliser la charge du système.

- a) le programme %B3 qui sera référencé par la suite "charge de type A".
- b) Le programme %M8: "charge de type B".
- c) Le programme %M9: "charge de type C".

Pour tester la réalité de ces charges, on les a surveillées à l'aide de la chaîne de programmes %L2, %L3, LB2, LBO, LB1, LB3, LB4.

Les résultats présentés dans les tableaux référencés TAB - 4.2.10, TAB - 4.2.11. et TAB - 4.2.12. ont été comparés avec les résultats de la surveillance à la Banque de Bruxelles qui sont considérées comme représentant la réalité. Ces résultats sont présentés et décrits en appendice A et les colonnes des tableaux référencés ci-dessus ont la même signification que celle des tableaux décrits dans l'appendice A.

C'est le programme %B3 qui semble le plus s'approcher de cette réalité. C'est le programme qui a été utilisé pour toutes les expériences nécessitant une charge du système.

4.2.2.1. On a effectué 5 exécutions du programme %B4 pour les valeurs (4,5,6,7,8 ticks) de la tranche de temps standard. La charge était constituée du programme %B3. Les résultats de ces expériences sont présentés en TAB - 4.2.1., TAB - 4.2.2., TAB - 4.2.3., TAB - 4.2.4. et TAB - 4.2.5. et sont illustrés par les graphiques référencés fig - 4.2.1., fig - 4.2.2., fig - 4.2.3., fig - 4.2.4. et fig - 4.2.5. On peut constater qu'il n'y a pratiquement pas de différence entre le type d'accès 3 et 4. De plus, on constate que pour la valeur 7 de la charge, la valeur du temps de réponse n'est pas modifiée quelque soit la valeur de la tranche de temps et quel que soit le type d'accès envisagé.

4.2.2.2. On a ensuite reconstitué ces tableaux de manière à mettre en évidence une éventuelle variation du

temps de réponse en fonction de la valeur de la tranche de temps. Le type d'accès devient ainsi le paramètre du tableau. Les résultats sont présentés dans TAB - 4.2.6., TAB - 4.2.7., TAB - 4.2.8. et TAB - 4.2.9. et sont illustrés par les graphiques référencés fig - 4.2.6., fig - 4.2.7., fig - 4.2.8. et fig - 4.2.9.

Ces figures montrent qu'il n'y a aucune relation entre le temps de réponse et la valeur de la tranche de temps standard.

- 4.2.2.3. On a enfin voulu voir les variations du temps de réponse en fonction du type de charge. On a fixé à 5 ticks la valeur de la tranche de temps standard et on a exécuté le programme %B4 pour les charges de type A, B et C. Les résultats de ces expériences sont présentés dans TAB - 4.2.13., TAB - 4.2.14. et TAB - 4.2.15. et sont illustrés par fig - 4.2.13., fig - 4.2.14., fig - 4.2.15., et fig - 4.2.16.



#### 4.3. Rôle de la méthode d'accès à la "data base" MUMPS-11.

La variable globale ↑ REP utilisé au paragraphe précédent était très simple. On peut dire, en effet, qu'à chaque niveau de la variable, il fallait accéder à un seul bloc disque. C'est le cas (la structure) optimum. En réalité, cette structure optimale est rarement atteinte.

De plus, on dispose de deux types de référence aux données d'une variable globale.

Dans ce paragraphe, on va tout d'abord présenter la structure logique et physique des variables globales. Les deux méthodes d'accès à une variable globale. Ensuite, on présentera un cas d'utilisation efficace de la référence nue [6], et ensuite une solution au problème de la concurrence pour l'accès à la "data base" MUMPS-11.

4.3.1. Dans ce paragraphe, on va décrire succinctement la structure des variables globales et leurs accès.

##### 4.3.1.1. Structure logique et physique des variables globales.

Le disque est divisé en zones physiques de stockage appelées blocs. Chaque bloc est spécifié par une zone adresse. Lorsqu'un bloc supplémentaire est nécessaire, un bloc continuation est alloué par le système. Ce bloc est aussi spécifié par une zone adresse vers lequel pointe le contenu du pointeur/continuation. Cette suite de blocs reliés entre-eux constitue une chaîne. Chaque niveau d'une variable réside dans une chaîne. (fig. 1,2)

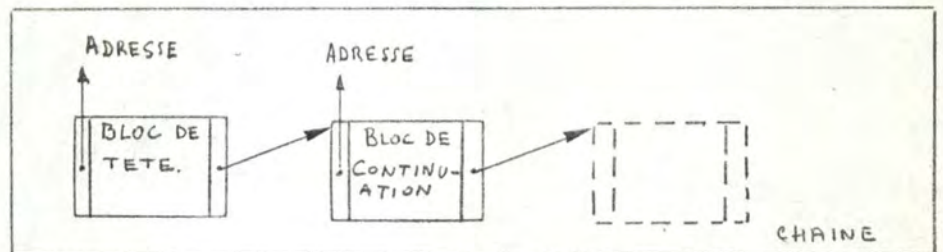


fig 1.

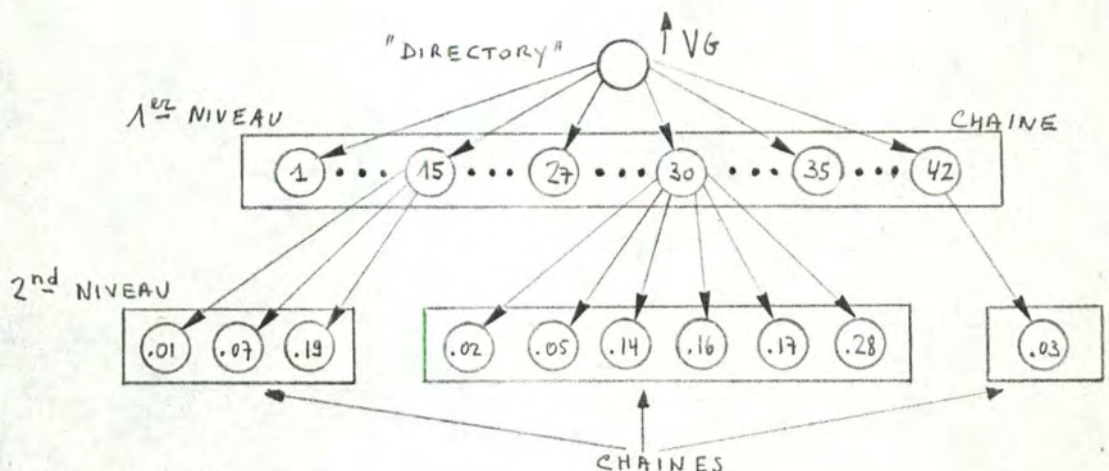


fig-2



La figure 2 montre l'organisation logique des données à chaque niveau. En réalité, à l'intérieur d'un bloc ou d'une chaîne de blocs, l'ordre logique et l'ordre physique ne coïncident pas nécessairement.

Cela ne constitue pas un inconvénient, le type d'accès est finalement décidé par le programmeur qui possède pour ce faire, 2 fonctions :

- a) `$H[IGH]` qui permet l'accès aux données dans l'ordre logique. (MUMPS-11 recherche dans la chaîne de blocs, l'indice qui juste au-dessus de celui spécifié en argument).
- b) `$Q[UERY]` qui permet l'accès aux données dans l'ordre physique. (MUMPS-11 renvoie simplement l'indice qui suit physiquement celui spécifié en argument).

Enfin, pour compléter l'organisation physique d'une variable globale sur disque, il reste à préciser 2 choses :

- a) A tous les niveaux autres que le plus bas, une entrée dans un bloc contient, outre la valeur de l'indice et d'éventuelles données, un pointeur dont le contenu est l'adresse du bloc adresse du bloc de tête de la chaîne correspondant au niveau immédiatement inférieur.
- b) L'information relative au niveau 0 d'une variable globale est contenue dans une entrée d'un bloc ou d'une chaîne de blocs, appelée "Directory". Le "directory" contient les informations relatives à toutes les variables globales.

En résumé, la figure 3 présente l'organisation physique associée à la structure logique présentée dans la figure 2.

#### 4.3.1.2. Accès physique des variables globales.

Le moyen d'accéder aux éléments d'une variable globale est différent selon que l'on utilise la référence globale ou la référence nue.

- a) Accès par l'intermédiaire de la référence globale.

Apriori, on peut dire que lorsque MUMPS-11 exécute une référence globale, il doit transférer dans la mémoire centrale au moins un bloc par niveau logique.

ex.: supposons une variable globale à 1 niveau. Supposons de plus qu'il y ait beaucoup de variables globales définies dans le système et que précisément, la variable globale envisagée soit dans le troisième bloc continuation du directory. Supposons aussi que cette variable globale ait 30 entrées tenant sur un



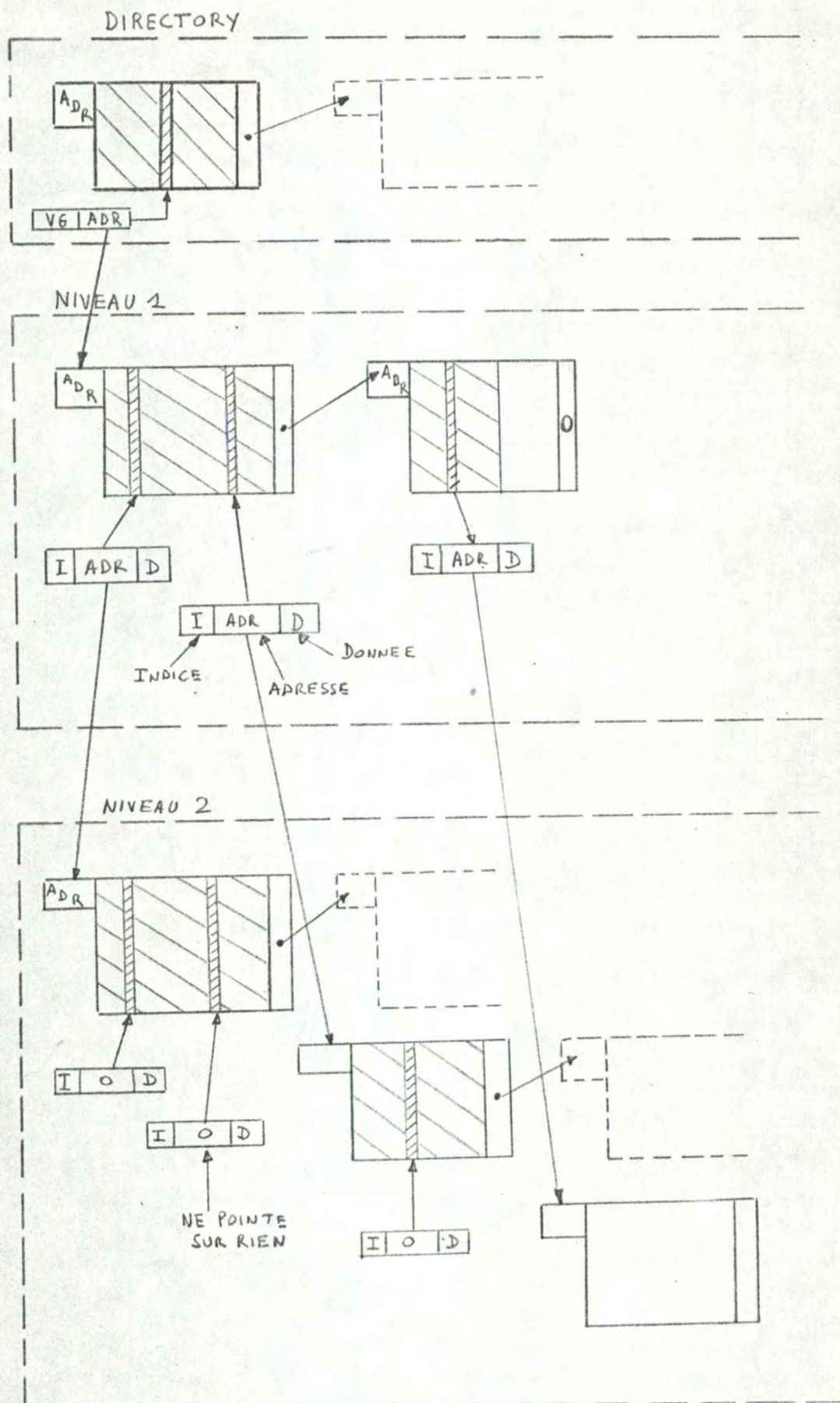


fig-3.



bloc et que 5 seulement soient intéressantes pour la suite des opérations. En utilisant la référence globale, il faudra réaliser 100 accès disques pour obtenir 5 données (en moyenne).

b) Accès par l'intermédiaire de la référence nue.

Rappelons qu'une référence nue doit toujours succéder à au moins une référence globale et que, lors de l'exécution de cette référence globale, un "Naked indicator" (voir appendice D) est rempli avec toutes les informations nécessaires à la définition complète d'une référence nue.

Physiquement, MUMPS-11 réalise cela au moyen de :

- a) l'adresse du bloc de tête du niveau atteint lors de la dernière référence globale.
- b) l'adresse du bloc qui était dans la mémoire centrale au moment de l'exécution de cette référence globale.

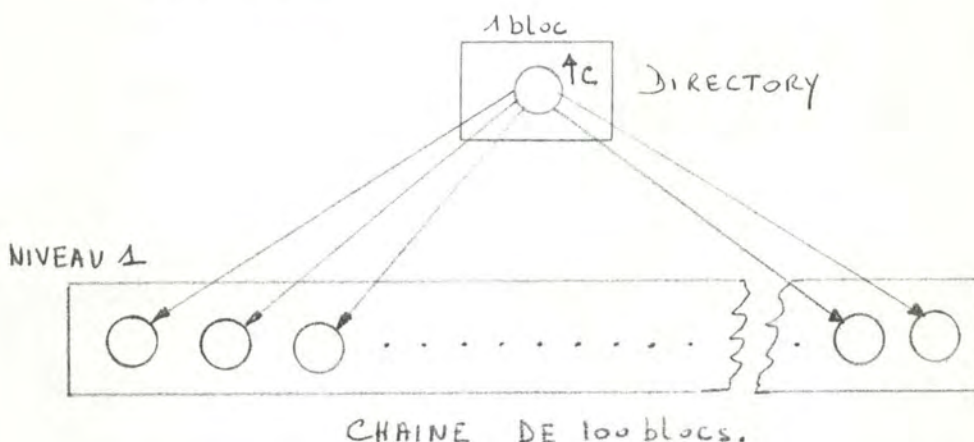
4.3.2. L'utilisation de la "Naked syntax" pour accéder à une variable globale relativement bien structurée peut permettre des améliorations du temps de réponse allant jusqu'à un facteur 10.

Prenons, pour exemple, la situation raisonnable de données de recensement relatif à 100 personnes dont 7 sont âgées de 30 ans.

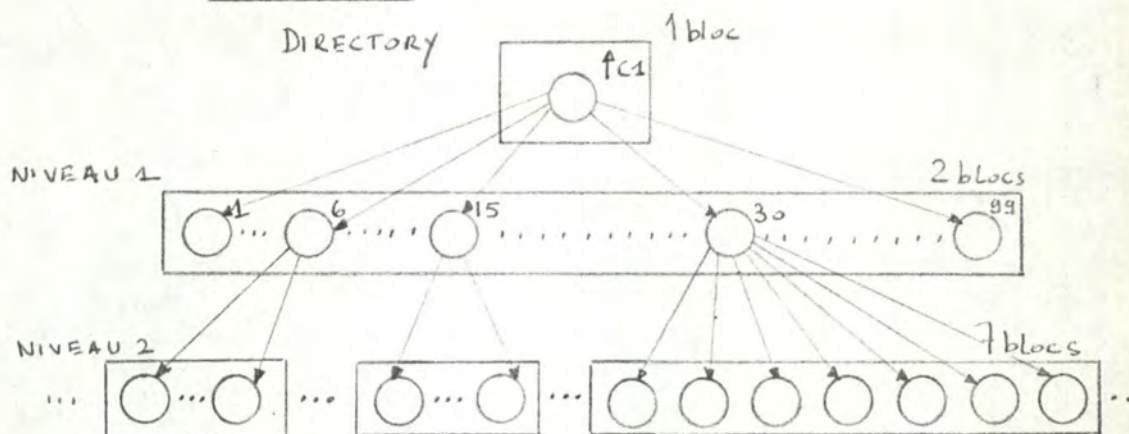
Supposons de plus que les données concernant une personne, tiennent sur un bloc.

- Problème : combien de blocs doivent être accédés pour obtenir les informations concernant la population âgée de 30 ans.
- En visageons deux manières possibles de structurer la variable globale.

Structure 1.





Structure 2.

Pour la structure 1, les données concernées sont réparties de façon équiprobables sur 101 blocs. Pour la structure 2, les données ne sont plus réparties que sur 10 blocs.

Si on effectue une référence nue à une donnée de la variable globale ayant la structure 1, on aura :

- 1 accès au bloc "directory".
- 1 recherche dans la chaîne de 100 blocs du niveau 1 dont on peut estimer qu'elle va coûter en moyenne  $\left[ \frac{100}{2} + .5 \right] = 50.5$  blocs à accéder.

Les informations relatives au 7% de la population âgée de 30 ans va ainsi coûter en moyenne

$$7 * \left[ \frac{100}{2} + .5 \right] + 1 = 354.5 \text{ blocs disques à accéder.}$$

Si on effectue une référence nue à une donnée dans la structure 2, on a :

- 1 accès au bloc du directory.
- 1.5 accès au bloc du directory.
- 1 recherche dans la chaîne de 7 blocs du niveau 2 ce qui, en moyenne, coûte

$$\left[ \frac{7}{2} + .5 \right] = 4 \text{ blocs à accéder.}$$

L'obtention des informations relatives aux 7% de la population âgée de 30 ans coûtera maintenant :

$$7 * \left[ \frac{7}{2} + .5 \right] + 1.5 + 1 = 30.5 \text{ blocs disque à}$$

accéder.

4.3.3. L'utilisation de la "Naked reference" n'est pas toujours sans danger dans l'implémentation du système MUMPS-11. La preuve est donnée par l'erreur fatale présentée dans l'appendice E (paragraphe B).

D'ailleurs, parallèlement à l'utilisation de la "Naked reference" (qui <sup>se</sup> présente plus comme une référence aveugle), se pose le problème de la concurrence et de l'ordre des utilisateurs pour la mise à jour des varia-

bles globales partageables de la "data base" MUMPS-11. En effet, le principe de Time-Sharing complique sérieusement le problème car un utilisateur qui a entamé la mise à zone d'éléments d'une variable globale et qui est interrompu en cours de traitement doit pouvoir retrouver intacts, les éléments sur lesquels il travaillait lorsqu'il reçoit à nouveau le contrôle du processeur. Ce délicat problème peut être surmonté par une convention de programmation pour les divers utilisateurs. On peut, en effet, résoudre cette question en utilisant les programmes des clés de protection dans les programmes. Une clé de protection est un "dummy device" que l'on assigne avant l'écriture, et qui est libéré une fois celle-ci terminée.

Les "dummy device" de MUMPS-11 sont les entrées 20 à 45 de la "Device Table" (voir appendice C) et qui sont référencées "Program Interlocks", dans le "MUMPS-11 Operator's Guide".

Ex.: A 21 S↑TST(1,2,3) = "JOB A" U21.



# RESULTATS DES MESURES DE TEMPS DE REPONSE.

A : Charge du système exprimée en nombre de partitions actives.  
B : Type d'accès au disque.

## 1. Tranche de temps de 4 ticks.

$\begin{smallmatrix} A \\ B \end{smallmatrix}$	1	2	3	4	5	6	7	8
1	22.44	25.48	31.04	41.32	54.08	72.76	90.08	95.04
2	24.48	32.04	36.12	42.88	60.56	82.60	95.00	112.52
3	27.36	32.04	39.36	48.56	63.32	83.96	94.88	116.24
4	27.28	34.40	44.76	57.96	62.60	89.08	95.36	118.60

TAB - 4.2.1.

## 2. Tranche de temps de 5 ticks.

$\begin{smallmatrix} A \\ B \end{smallmatrix}$	1	2	3	4	5	6	7	8
1	18.68	26.12	32.20	44.56	55.20	73.76	93.36	95.84
2	22.36	28.60	42.20	46.96	63.96	78.52	95.92	107.08
3	26.04	32.44	39.60	51.16	65.16	88.80	94.96	114.72
4	27.88	34.76	40.24	53.24	63.96	92.60	96.72	122.00

TAB - 4.2.2.

## 3. Tranche de temps de 6 ticks.

$\begin{smallmatrix} A \\ B \end{smallmatrix}$	1	2	3	4	5	6	7	8
1	18.76	25.40	35.40	41.60	55.16	71.36	95.36	94.60
2	22.44	30.56	40.44	49.36	60.96	79.56	95.60	108.76
3	26.04	34.28	39.68	55.44	61.72	87.08	95.04	120.08
4	27.88	33.12	42.44	48.80	66.92	84.48	96.88	121.36

TAB - 4.2.3.

## 4. Tranche de temps de 7 ticks.

$\begin{smallmatrix} A \\ B \end{smallmatrix}$	1	2	3	4	5	6	7	8
1	18.28	24.64	32.52	44.04	58.68	73.16	95.16	96.12
2	24.44	30.56	39.24	47.16	61.00	78.96	94.12	111.60
3	25.44	31.24	40.64	49.28	60.12	83.92	94.88	122.64
4	27.12	32.80	42.80	55.76	64.52	84.12	94.44	122.44

TAB - 4.2.4.

5. Tranche de temps de 8 ticks.

$\begin{smallmatrix} A \\ B \end{smallmatrix}$	1	2	3	4	5	6	7	8
1	20.36	25.36	36.20	45.24	52.64	74.76	93.16	95.36
2	24.56	35.04	36.92	44.00	63.24	78.52	94.28	108.48
3	27.48	33.88	39.72	48.32	62.92	90.44	95.16	122.00
4	27.24	35.44	42.40	52.40	64.36	83.52	94.36	123.76

TAB - 4.2.5.

+ + + + + + + +

A : Charge du système exprimée en nombre de partitions actives.  
 B : Valeur en nombre de ticks d'une tranche de temps standard.

1. Type d'accès 1.

$\begin{smallmatrix} A \\ B \end{smallmatrix}$	1	2	3	4	5	6	7	8
4	22.44	25.48	31.04	41.32	54.08	72.76	90.08	95.04
5	16.68	26.12	32.20	44.56	55.20	73.76	93.36	95.84
6	18.76	25.40	35.40	41.60	55.16	71.36	95.36	94.60
7	18.28	24.64	32.52	44.04	58.68	73.16	95.16	96.12
8	20.36	25.36	36.20	45.24	52.64	74.76	93.16	95.36

TAB - 4.2.6.

2. Type d'accès 2.

$\begin{smallmatrix} A \\ B \end{smallmatrix}$	1	2	3	4	5	6	7	8
4	24.48	32.04	36.12	42.88	60.56	82.60	95.00	112.52
5	22.36	28.60	42.20	46.96	63.96	78.52	95.92	107.08
6	22.44	30.56	40.44	49.36	60.96	79.56	95.60	108.76
7	24.44	30.56	39.24	47.16	61.00	78.96	94.12	111.60
8	24.56	35.04	36.92	44.00	63.24	78.52	94.28	108.48

TAB - 4.2.7.



3. Type d'accès 3.

$\begin{smallmatrix} A \\ B \end{smallmatrix}$	1	2	3	4	5	6	7	8
4	27.36	32.04	39.36	48.56	63.32	83.96	94.88	116.24
5	26.04	32.44	39.60	51.44	65.16	88.80	94.96	114.72
6	26.04	43.28	39.68	55.44	61.72	87.08	95.04	120.08
7	25.44	31.24	40.64	49.28	60.12	83.92	94.88	122.64
8	27.48	33.88	39.72	48.32	62.92	90.44	95.16	122.00

TAB - 4.2.8.

4. Type d'accès 4.

$\begin{smallmatrix} A \\ B \end{smallmatrix}$	1	2	3	4	5	6	7	8
4	27.28	43.40	44.76	57.96	62.60	89.08	95.36	118.60
5	27.88	34.76	40.24	53.24	63.96	92.60	96.72	122.00
6	27.88	33.12	42.44	48.80	66.92	84.48	96.88	121.36
7	27.12	32.80	42.80	55.76	64.52	89.12	94.44	122.44
8	27.24	35.44	42.40	52.40	64.36	83.52	94.36	123.76

TAB - 4.2.9.

TAB - 4.2.10

ETAT	1	2	3	4	5	6	7	8		
28	/	/	/	/	/	/	/	/	/	/
29	/	/	/	/	/	/	/	/	/	/
30	/	/	/	/	/	/	/	/	/	/
32	18.21	20.15	17.05	23.64	17.44	/	15.89	15.89	16.03	1.28
41	/	/	/	/	/	100	/	/	12.50	1.00
42	1.55	1.16	0.77	1.16	1.55	/	2.32	1.55	1.25	0.10
43	16.66	22.86	24.41	26.74	29.06	/	27.51	26.35	21.70	1.73
44	10.46	11.62	8.52	6.97	8.91	/	7.36	7.75	7.70	0.61
45	/	/	/	/	/	/	/	/	/	/
46	/	/	/	/	/	/	/	/	/	/
47	/	/	/	/	/	/	/	/	/	/
48	1.93	1.55	1.55	1.16	1.93	/	1.55	4.26	1.74	0.13
49	/	/	/	/	/	/	/	/	/	/
50	51.16	42.63	47.67	40.31	41.08	/	45.34	44.18	39.05	3.12
51	/	/	/	/	/	/	/	/	/	/

CHARGE DE TYPE A

INTERVALLE DE MONITORING : 10 SECONDES.  
 DUREE DU MONITORING : 2580 SECONDES (258 EVENEMENTS) (43 minutes).  
 NOMBRE DE JOBS ACTIFS : 8  
 GARBAGE COLLECTOR : 20



TAB - 4.2.11

ETAT	1	2	3	4	5	6	7	8		
28	/	/	/	/	/	/	/	/	/	/
29	/	/	/	/	/	/	/	/	/	/
30	/	/	/	/	/	/	/	/	/	/
32	13.17	12.57	9.58	14.97	9.58	/	14.37	13.17	10.92	0.87
41	/	/	/	/	/	100	/	/	12.50	1.00
42	1.79	/	0.59	/	/	/	0.59	/	0.37	0.02
43	10.17	7.78	12.57	14.97	7.18	/	5.38	9.58	8.45	0.67
44	1.19	2.99	1.19	1.19	2.39	/	2.99	1.19	1.64	0.13
45	/	/	/	/	/	/	/	/	/	/
46	/	/	/	/	/	/	/	/	/	/
47	0.59	0.59	0.59	0.59	0.59	/	0.59	0.59	0.52	0.04
48	21.55	20.35	20.35	19.16	25.14	/	19.76	22.15	18.56	1.48
49	/	/	/	/	/	/	/	/	/	/
50	51.49	55.68	55.08	49.10	55.08	/	56.28	53.29	47.00	3.76
51	/	/	/	/	/	/	/	/	/	/

CHARGE DE TYPE B

INTERVALLE DE MONITORING : 10 SECONDES  
DUREE DE MONITORING : 1670 SECONDES (167 EVENEMENTS) (28 minutes).  
NOMBRE DE JOBS ACTIFS : 8  
GARBAGE COLLECTOR : 0



TAB - 4.2.12.

ETAT	1	2	3	4	5	6	7	8		
28	/	/	/	/	/	/	/	/	/	/
29	/	/	/	/	/	/	/	/	/	/
30	/	/	/	/	/	/	/	/	/	/
32	11.33	6.66	/	4.66	7.33	8.66	10.66	10.66	7.50	0.60
41	/	/	100	/	/	/	/	/	12.50	1.00
42	/	/	/	/	/	/	/	/	/	/
43	5.33	9.33	/	12.66	8.00	5.33	7.33	4.00	6.50	0.52
44	2.66	2.00	/	2.66	2.66	2.00	2.00	2.66	2.08	0.16
45	/	/	/	/	/	/	/	/	/	/
46	/	/	/	/	/	/	/	/	/	/
47	/	/	/	/	/	/	/	/	/	/
48	6.66	4.00	/	7.33	7.33	7.33	7.33	6.00	5.75	0.46
49	/	/	/	/	/	/	/	/	/	/
50	74.00	78.00	/	72.66	74.66	76.66	72.66	76.66	65.66	5.25
51	/	/	/	/	/	/	/	/	/	/

INTERVALLE DE MONITORING : 10 SECONDES  
 DUREE DU MONITORING : 1500 SECONDES (150 EVENEMENTS) (25 minutes)  
 NOMBRE DE JOBS ACTIFS : 8  
 GARBAGE COLLECTOR : 0

CHARGE DE TYPE C.



A : Charge du système exprimée en nombre de partitions actives.  
B : type d'accès au disque

1. CHARGE DE TYPE A.

$\begin{smallmatrix} A \\ B \end{smallmatrix}$	1	2	3	4	5	6	7	8
1	18.68	26.12	32.20	44.56	55.20	73.76	93.36	95.84
2	22.36	28.60	42.20	46.96	63.96	78.52	95.92	107.08
3	26.04	32.44	39.60	51.44	65.16	88.80	94.96	114.72
4	27.88	34.76	40.24	53.24	63.96	92.60	96.72	122.00

TAB - 4.2.13.

2. CHARGE DE TYPE B.

$\begin{smallmatrix} A \\ B \end{smallmatrix}$	1	2	3	4	5	6	7	8
1	19.46	23.17	25.17	29.14	32.06	35.28	46.00	50.46
2	24.50	29.22	33.80	37.97	42.03	44.68	46.50	56.40
3	25.60	32.24	35.05	37.34	44.30	49.28	55.70	62.86
4	27.82	34.11	37.00	39.65	43.90	51.63	56.35	63.06

TAB - 4;2.14.

3. CHARGE DE TYPE C.

$\begin{smallmatrix} A \\ B \end{smallmatrix}$	1	2	3	4	5	6	7	8
1	18.34	21.28	22.85	27.71	32.56	30.16	34.45	42.33
2	24.92	25.57	29.92	32.65	34.10	40.12	41.65	40.46
3	25.04	27.88	32.10	32.85	36.60	41.88	46.35	47.20
4	27.22	30.24	33.72	38.34	40.06	46.52	50.90	55.06

TAB - 4.2.15.

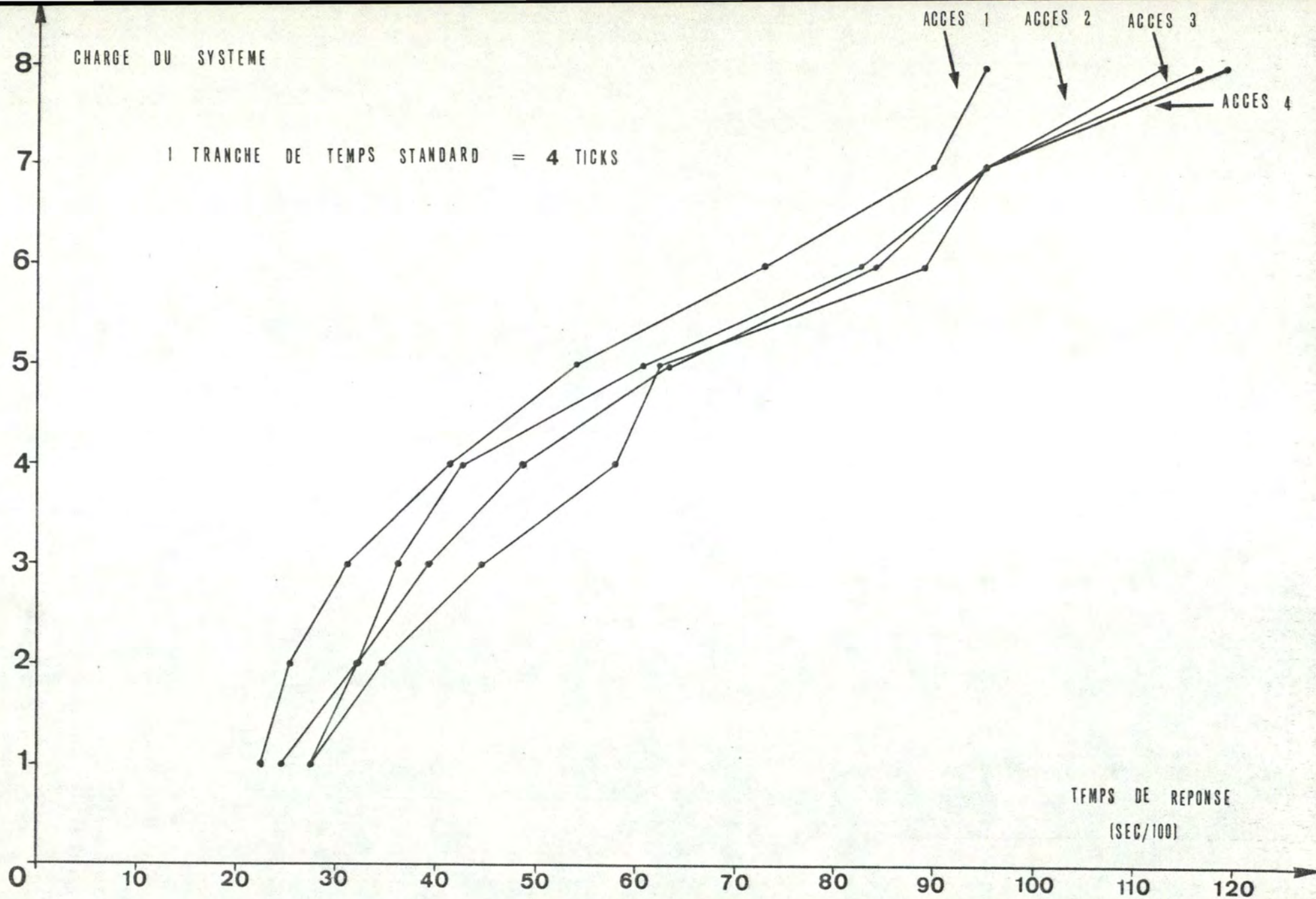


FIG-4.2.1



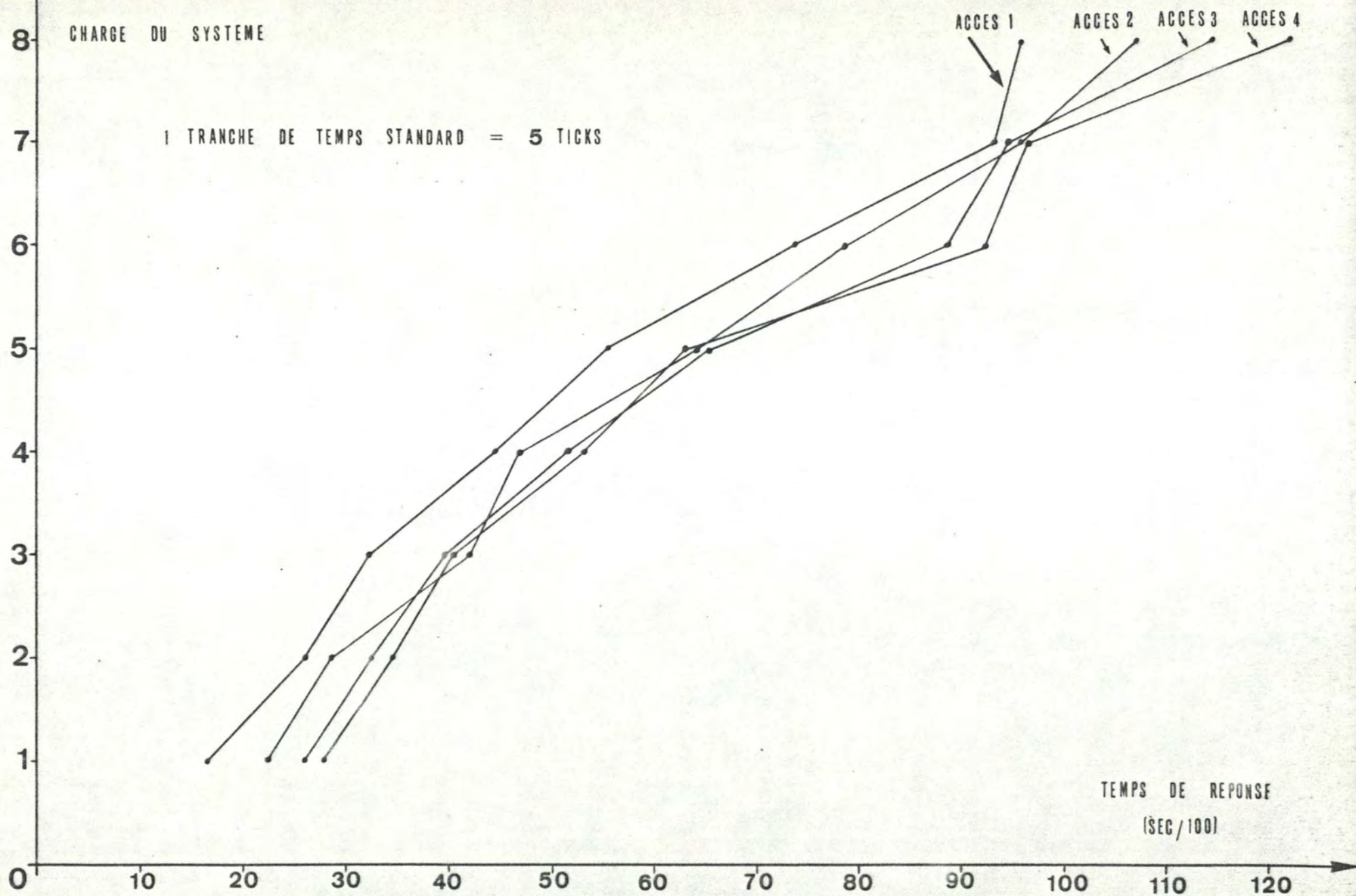


FIG-4.2.2

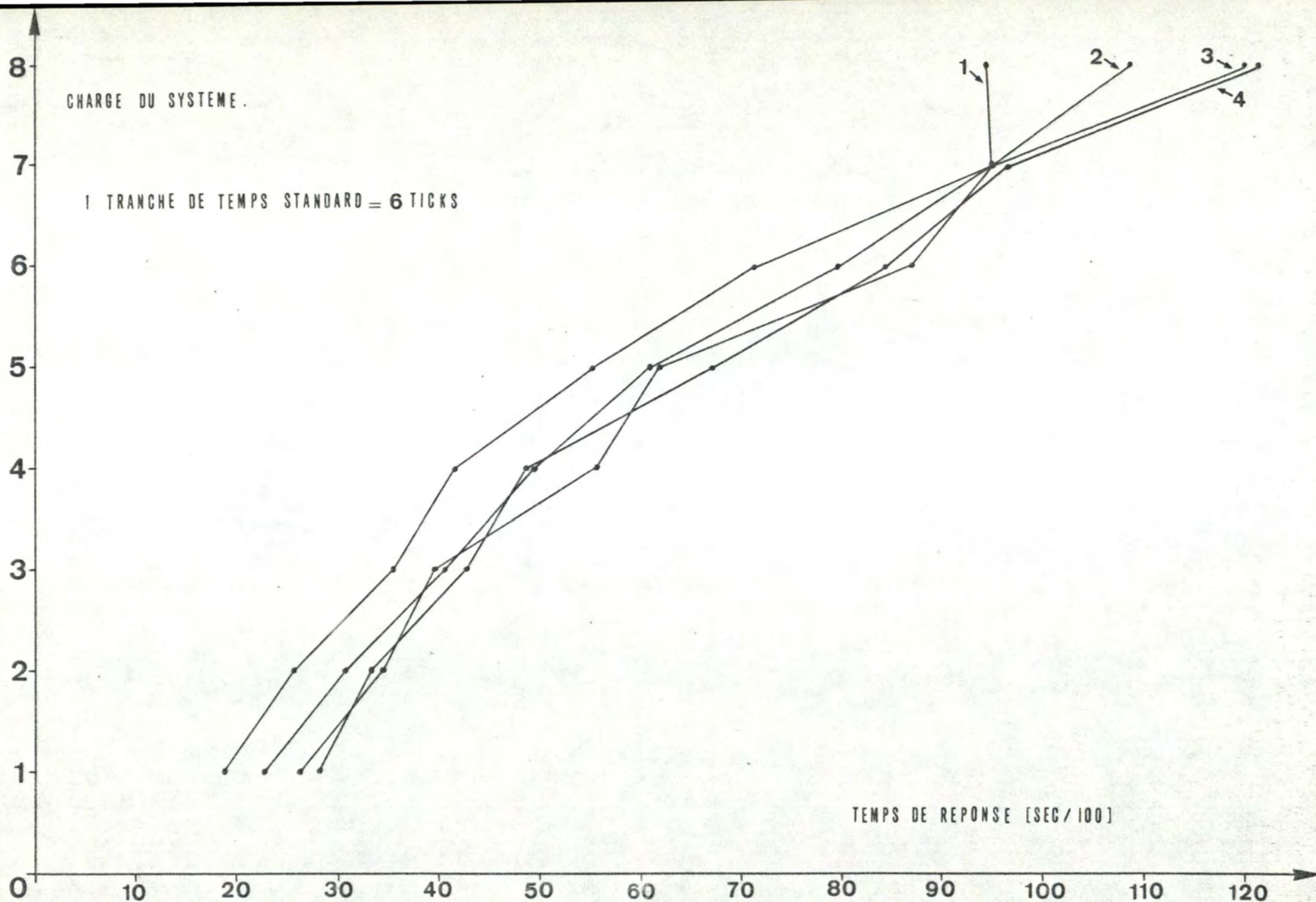


FIG-4.2.3



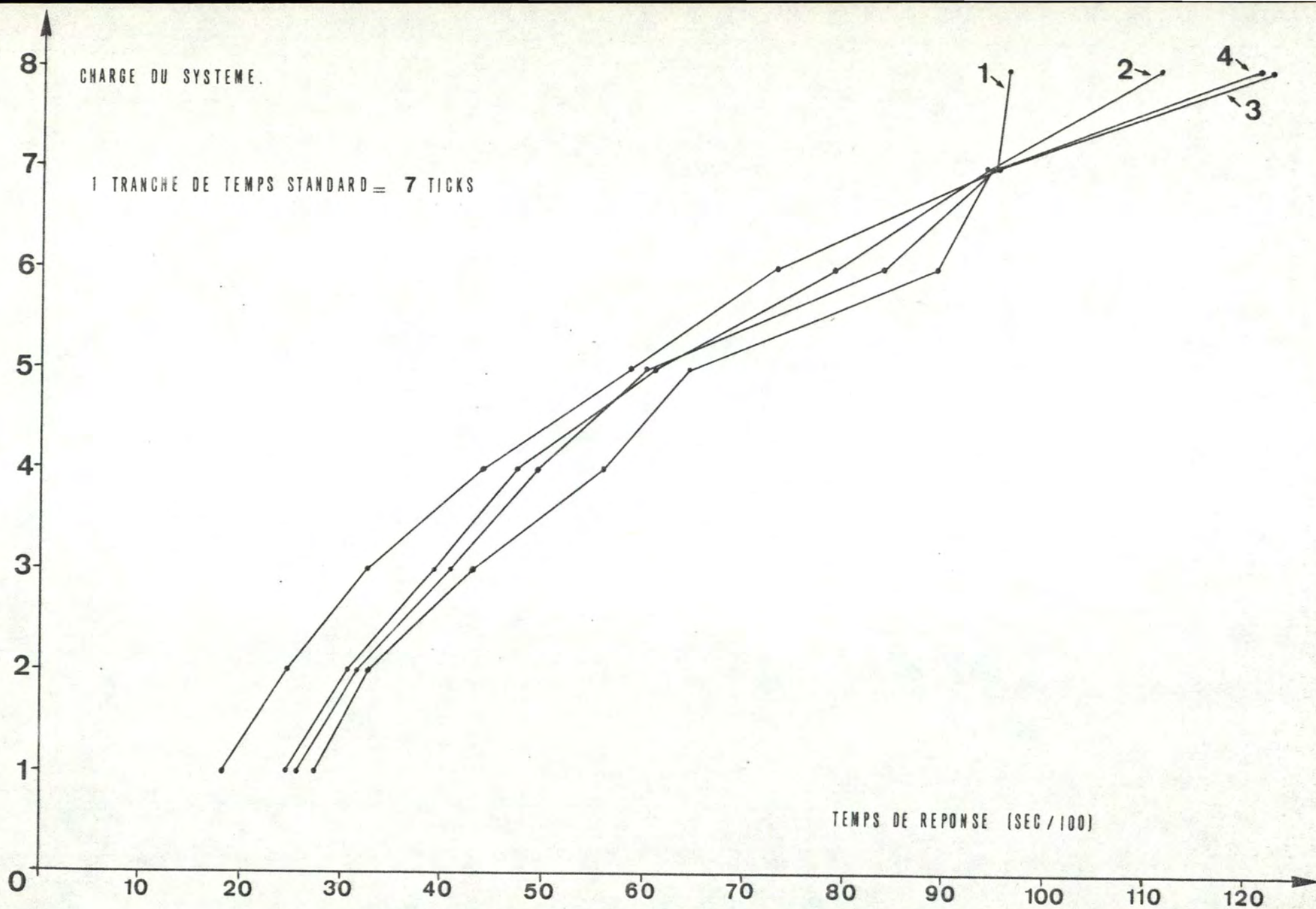


FIG-4.2.4

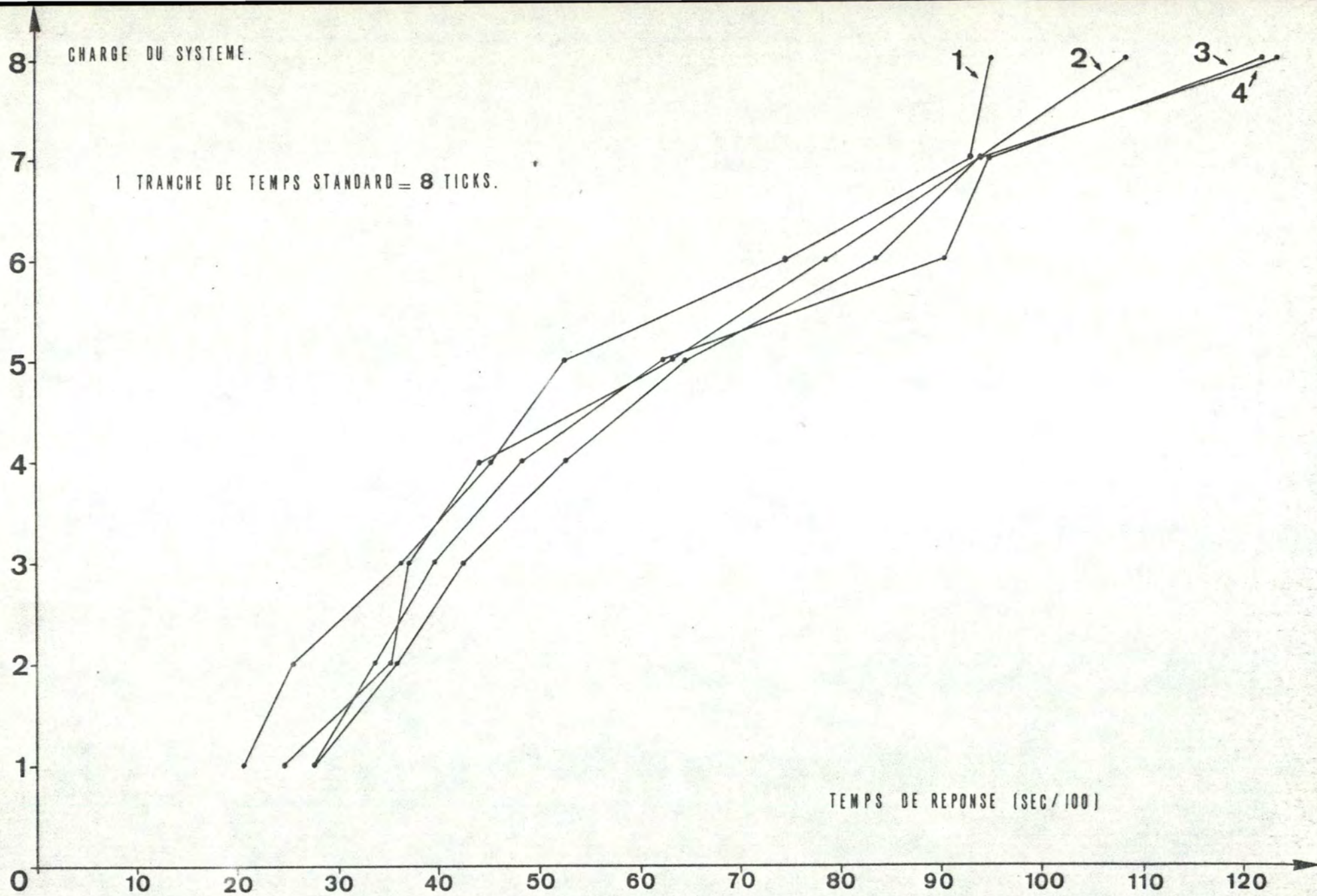


FIG-4.2.5



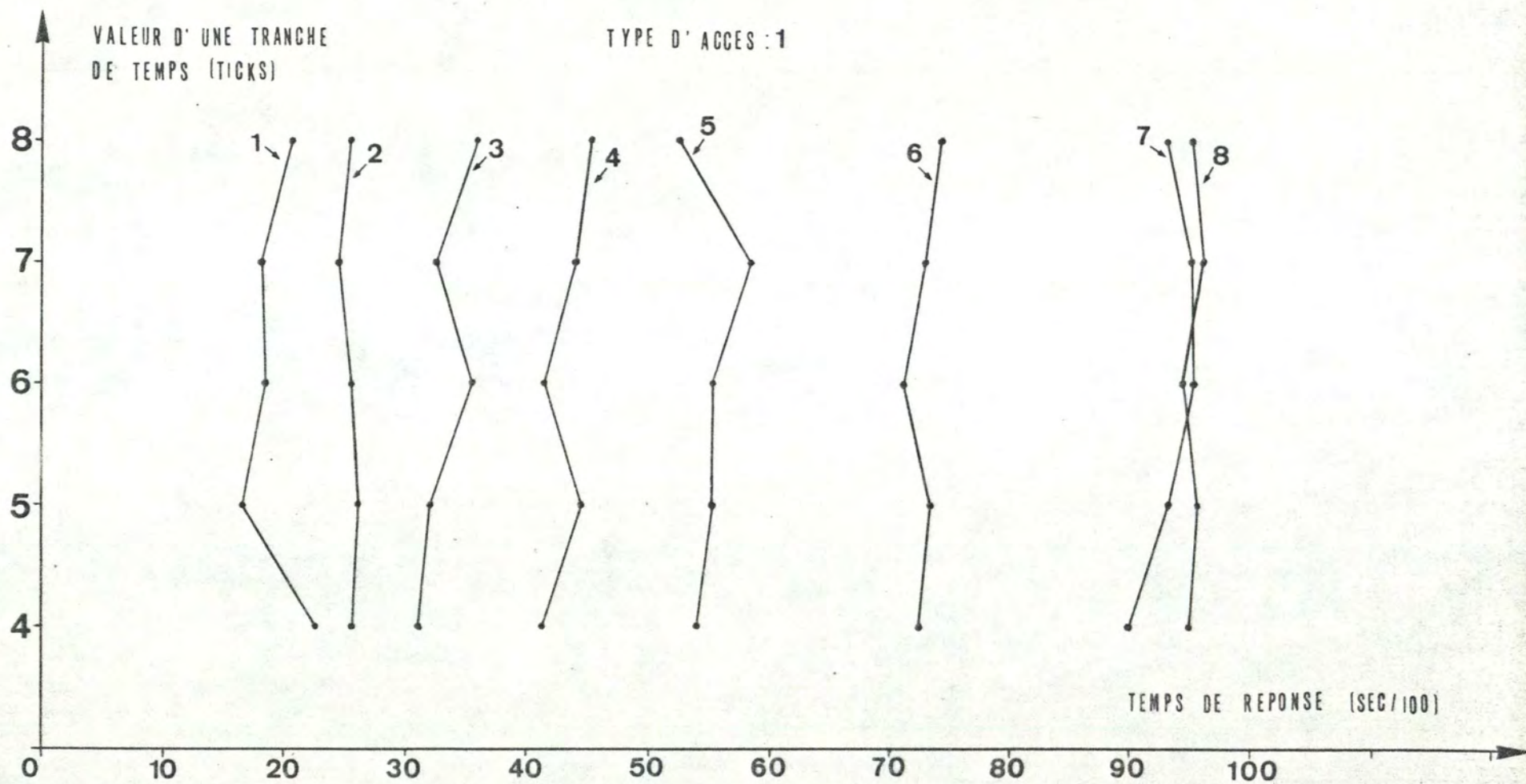


FIG-4.2.6

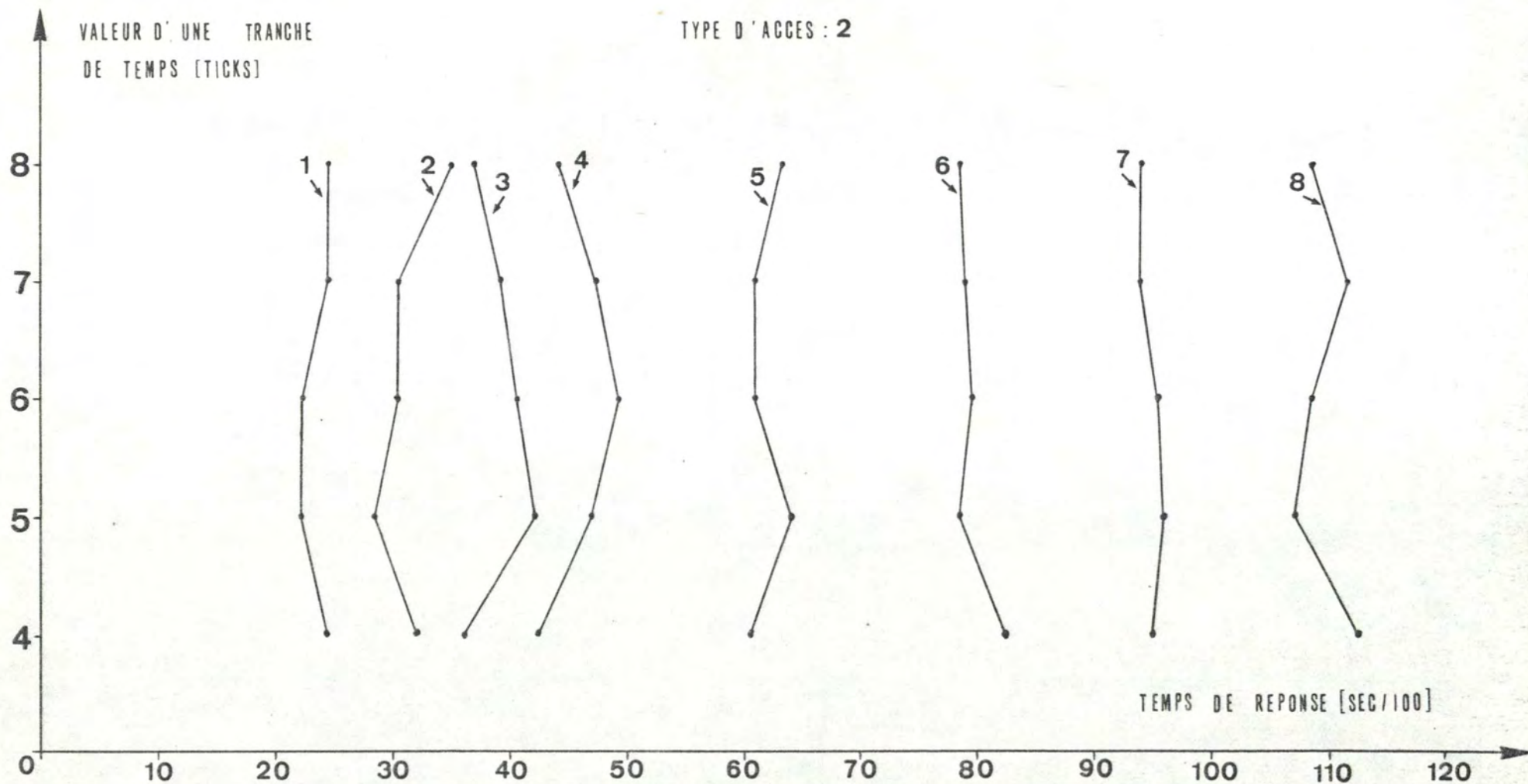


FIG-4.2.7



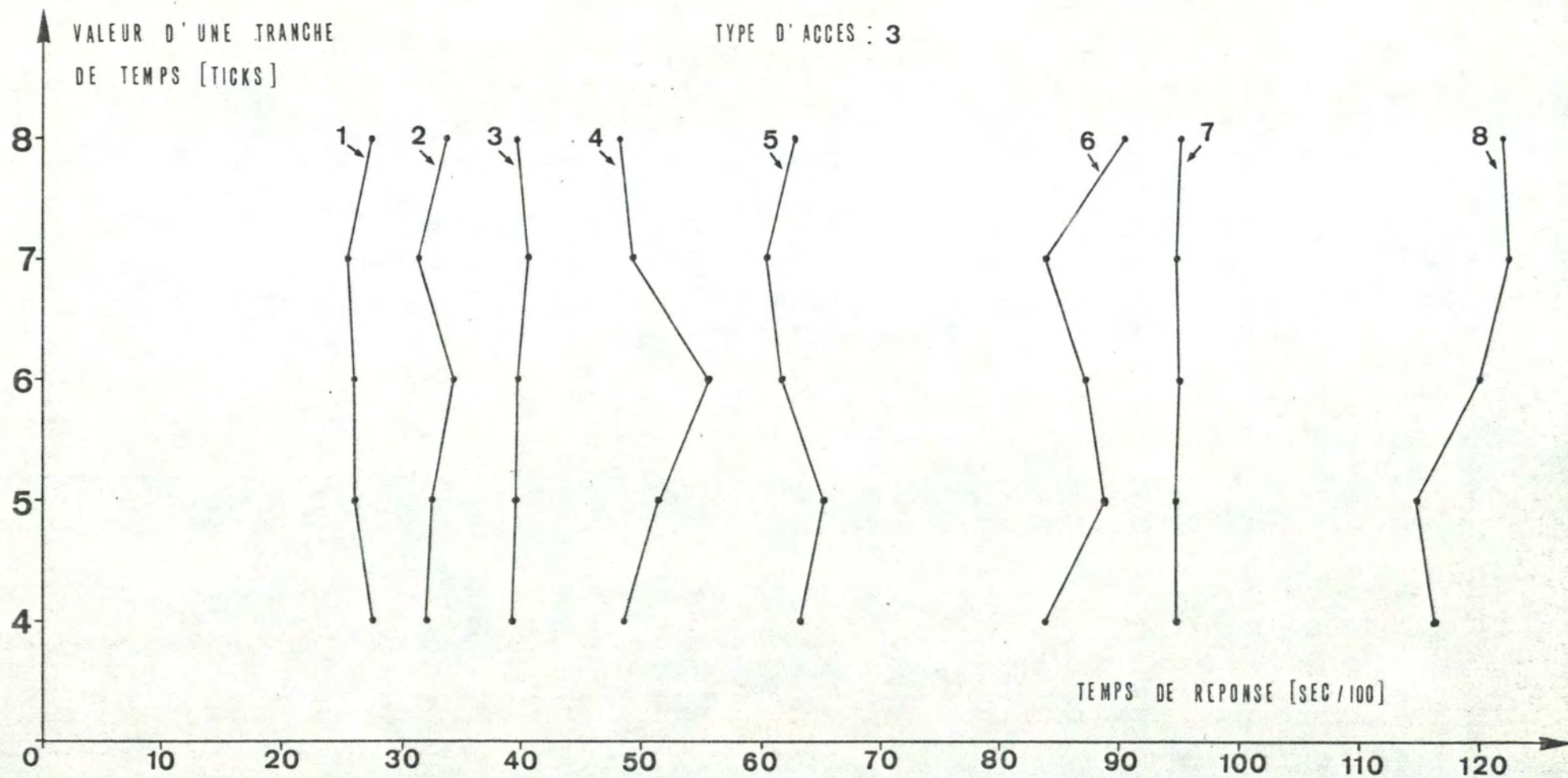


FIG-4.2.8

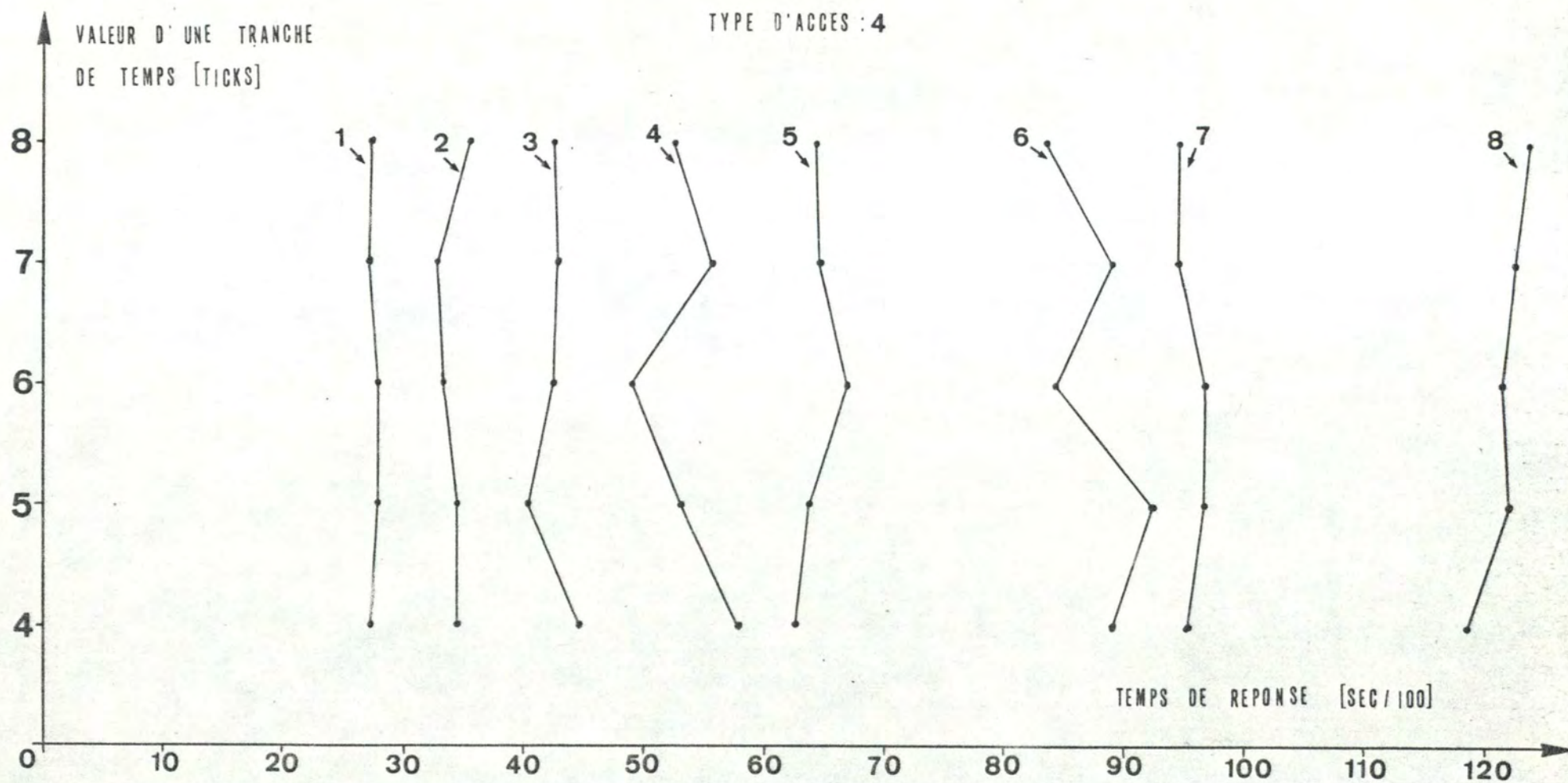


FIG-4.2.9



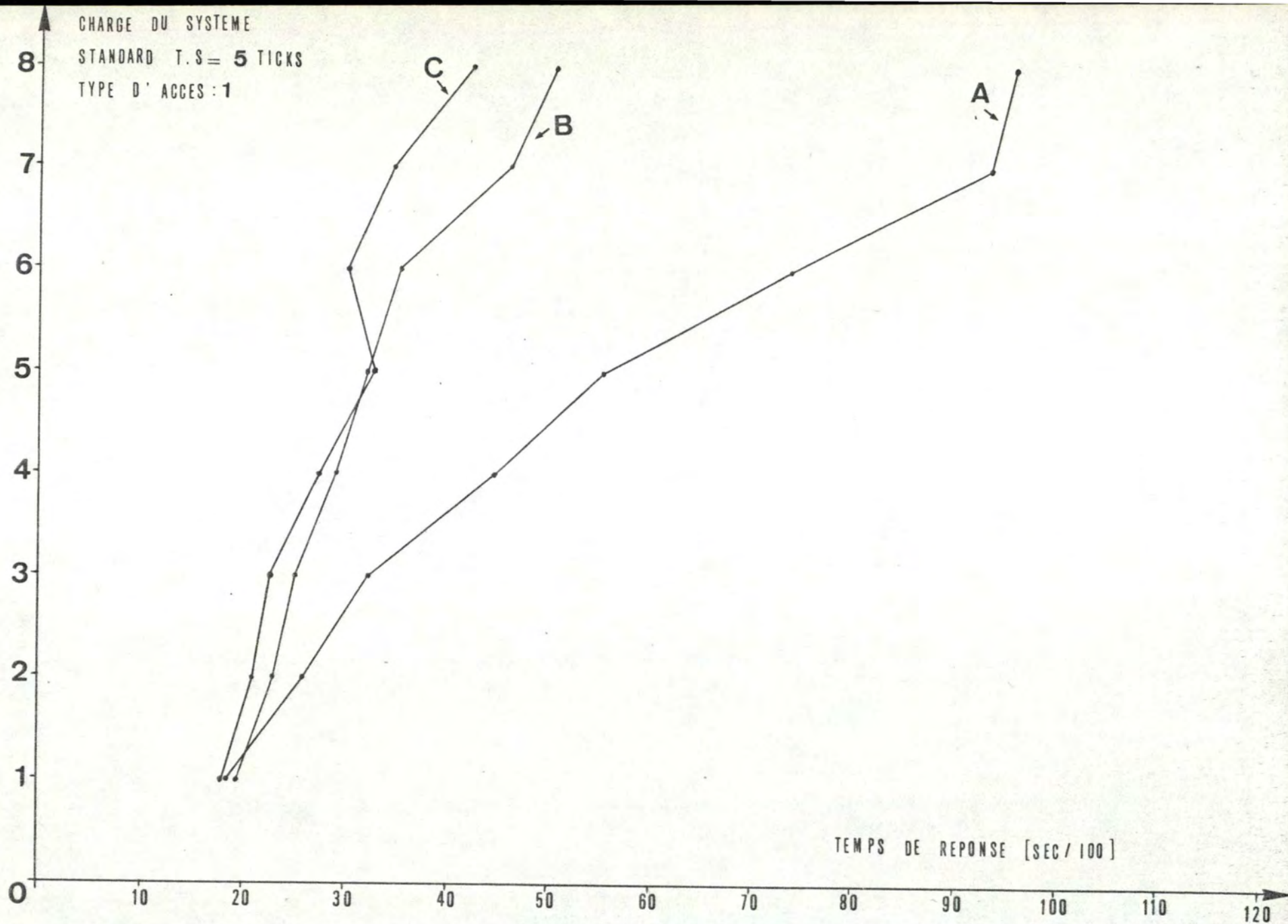


FIG-4.2.13

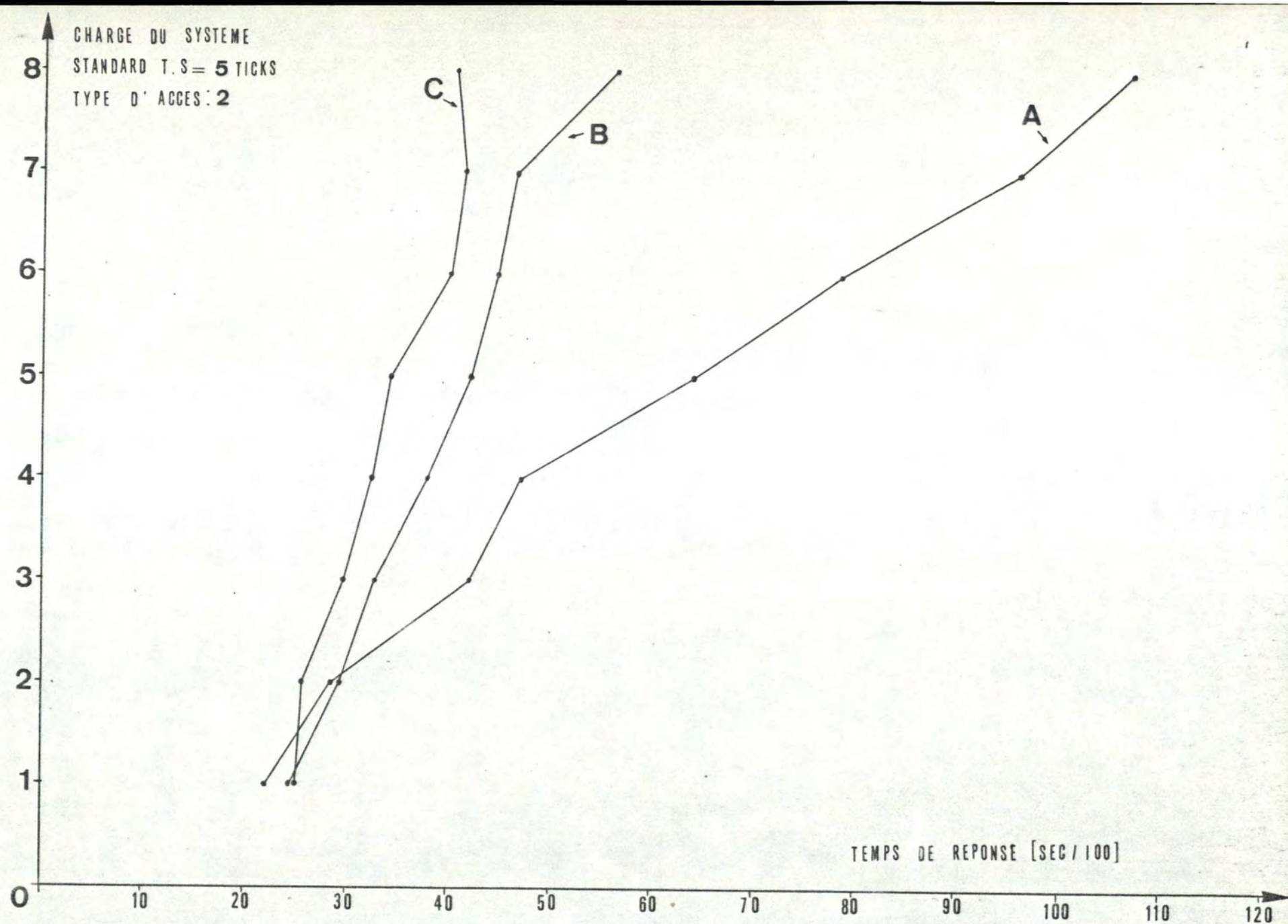


FIG-4.2.14



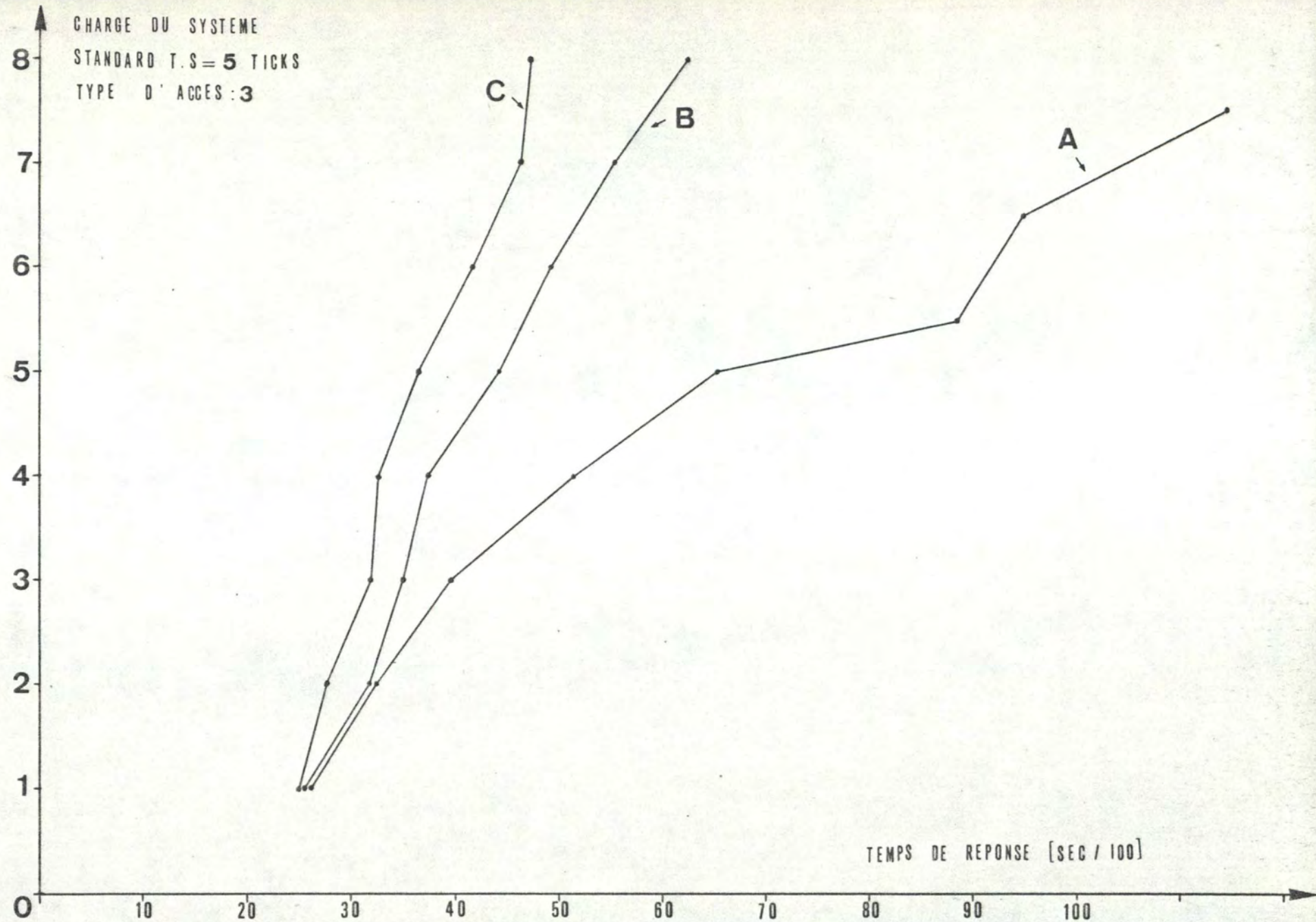


FIG-4.2.15

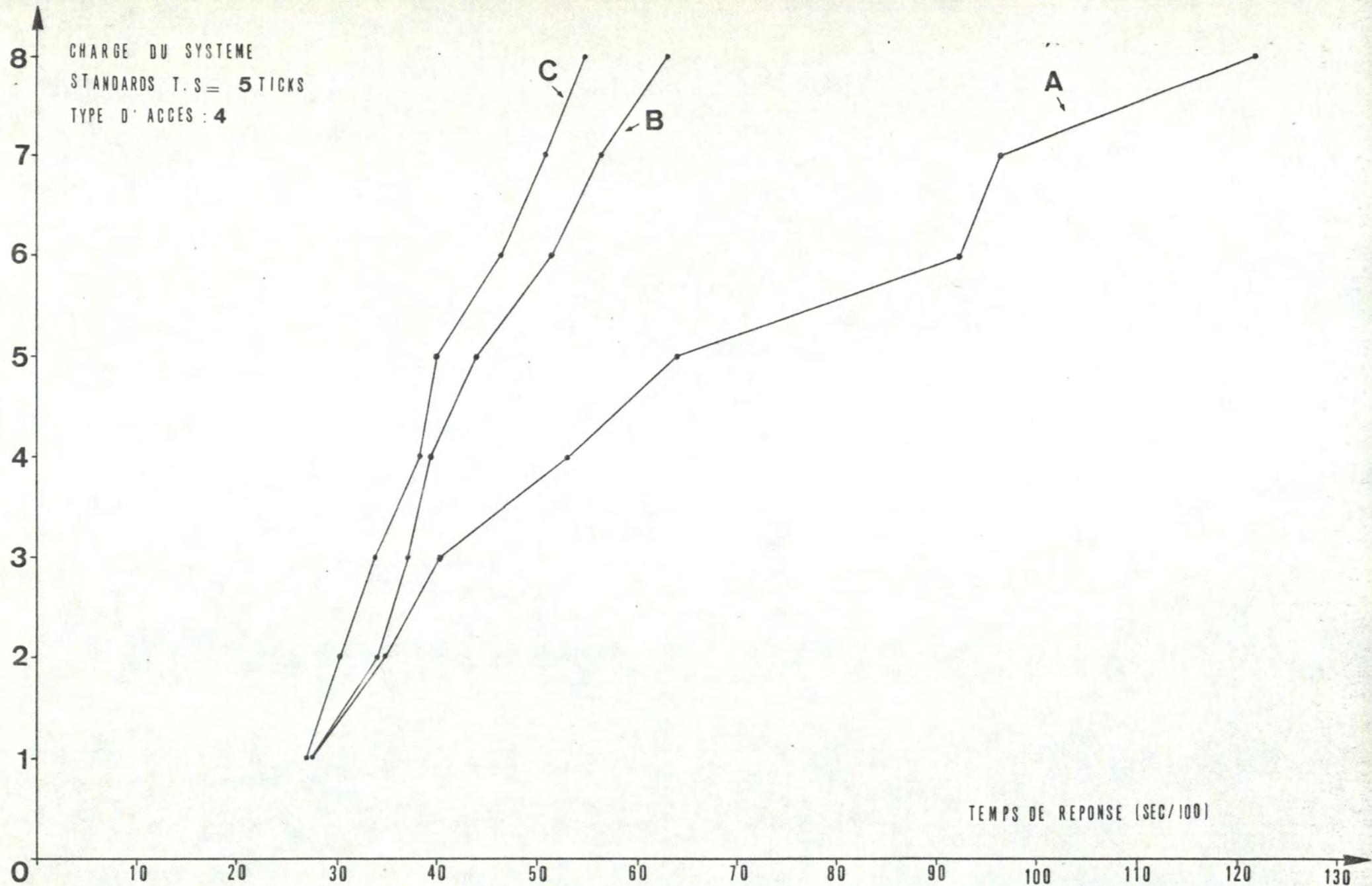


FIG-4.2.16



## C O N C L U S I O N S .

- . Le travail qui vient d'être réalisé sur un cas concret, l'a été de manière à mettre en valeur le point suivant : "The key to performance evaluation as well as to systems design is an understanding of what systems are and how they work" [21].

Les conclusions techniques du chapitre II et des différents paragraphes du chapitre III, ainsi que le contenu de l'appendice E en sont autant d'illustrations.

Etant donné que MUMPS-11 n'a manifestement pas été conçu et dessiné pour permettre la mesure de sa performance, l'éventail des mesures réalisables est relativement modeste. Toutefois, les mesures réalisées sont originales. De plus, l'analyse du programme de surveillance, outre qu'il a soulevé pas mal de problèmes, a permis de mettre en lumière les interactions existantes entre un système d'exploitation et un programme d'application.

- . A côté des appréciations techniques présentées au fur et à mesure des expériences effectuées dans le cadre de l'évaluation de performance, la réalisation de ce mémoire a également permis de dégager un certain nombre d'avantages et de limitations de MUMPS-11.

Au rayon des avantages, on citera notamment:

- . la souplesse et la facilité des fonctions destinées à la manipulation des "strings".
- . la programmation modulaire favorisée par la structure du langage.
- . les "records" de longueur variable.
- . l'allocation dynamique de zones de stockage tant sur disque qu'en mémoire centrale, qui réalise l'implémentation du principe suivant : "data takes only the space that is strictly necessary".

Quant aux limitations de MUMPS-11, elles peuvent se répartir comme suit:

- . la vitesse d'exécution d'un programme interprétatif réalisant un traitement essentiellement CPU peut être dégradée d'un facteur 20 à 1 par rapport au code correspondant généré par un compilateur.  
Toutefois, peu de programmes font du traitement "CPU bound" en mode "on line" dans un contexte de gestion de données. Dans un tel contexte, la partie importante d'un travail est consacrée au traitement de textes et à la manipulation du disque. Dans les langages assembleur ou nécessitant l'emploi d'un compilateur, de telles activités sont implémentées par l'intermédiaire d'une séquence d'appels à des routines spécialisées, et dès lors l'utilisation d'un interpréteur comme moyen de générer ces appels aux routines spécialisées plutôt que la compilation de ces appels entraîne seulement une petite quantité d'"overhead processing".
- . il n'existe aucun "interlock" hardware ou software pour l'accès à la "data base".
- . la longueur maximum d'un "string" est fixée à 132 caractère.



Cela étant, l'impression générale qui prévaut lorsqu'on a eu l'occasion de travailler avec MUMPS-11, est celle de facilité en ce qui concerne la mise au point de programme d'applications et celle de souplesse en ce qui concerne la manipulation de textes et de variables globales.

+ + + + +  
+ + + +  
+ + +  
+ +  
+



## APPENDICE.A

Présentation des informations et résultats obtenus à la Banque de Bruxelles.

A. Cette collecte peut se résumer comme suit :

- A.1. Le programme d'échantillonnage sans autre forme de commentaire que la signification du contenu de ses instructions.
- A.2. Une sortie sur imprimante de la surveillance de la journée du 18/02/75. Un exemple d'une telle sortie est présenté dans le tableau 1.
- A.3. Les tableaux 2,3,4 présentent des moyennes et pourcentages calculés à partir du monitoring du 14/02/75 et 18/02/75.

B. Description du tableau 1.

- B.1. Les différentes informations contenues dans une ligne de monitoring sont séparées par la délimitation "\*".
  - colonne 1 : temps présenté sous forme "heure:minute"
  - colonne 2 : nombre de jobs actifs dans le système.
  - colonne 3 17 : indiquent l'état des partitions  
1 → 14.
  - colonne 18 : état de l'imprimante (device # 3).
  - colonne 19 : état du VIEW Device # 46 (View Memory Only).
  - colonne 20 : état de la bande magnétique (Device # 47).
  - colonne 21 : état du View Device # 63 (View disk or memory).
- B.2. Signification des codes utilisés dans les colonnes 3 - 17.
  - 28 : "I/O hung terminal"
  - 29 : "I/O hung dectape"
  - 30 : "I/O hung magtape"
  - 32 : " transition state"
  - 41 : En "Run queue"
  - 42 : En "short queue"
  - 43 : En "disk I/O bound queue"
  - 44 : En "disk resource bound queue"
  - 45 : En "ring buffer resource bound queue"
  - 46 : En "256 - word buffer resource bound queue"
  - 47 : En "high priority wait queue"
  - 48 : En "medium priority wait queue"
  - 49 : En "low priority wait queue"
  - 50 : En "clock queue"
  - 51 : En "partition available queue"
- B.3. Convention pour l'état des devices 3,46,47,63.  
 Si "oo" alors device inoccupé.  
 Si différent de "oo" alors on trouve le numéro de la



partition à laquelle est assigné le device.

### C. Description du tableau 2.

Ce tableau concerne la journée du 14/02/75.

- C.1. La colonne 0 contient les différents états possibles dans lesquels on peut trouver une partition.
- C.2. Les colonnes 1 - 14 contiennent pour chaque état le pourcentage des cas où on a trouvé la partition dans cet état.  
Certaines partitions ont eu des tâches très spécifiques à effectuer, ainsi :
- les partitions 2/7/9/10/11 ont réalisés des tâches interactives (saisie de données).
  - La partition 8 a, quant à elle, réalisé des tâches plus orientées "batch processing mode".
  - La partition 12 a traité le programme de monitoring.
- C.3. Dans la colonne 15, l'auteur du programme a voulu montrer comment se classaient les divers états, les uns par rapport aux autres, cela, hormis l'état 51 (partition disponible) et en essayant de masquer globalement l'influence de la partition qui exécutait le programme de monitoring.  
Le pourcentage calculé à cet effet, l'a été de la manière suivante :

Nombre total des occurrences d'un état

$$\frac{[(\text{Nbre de partitions}-1) \cdot \text{Nbre tot. des occurrences de l'état 51}]}{100}$$

En fait, cela revient à faire la somme des éléments d'une ligne pour les colonnes 1 - 14 du tableau et à diviser cette somme par le nombre moyen de jobs actifs d'une unité.

- C.4. Dans la colonne 16, on présente le nombre moyen de partitions que l'on trouve pour chaque état excepté l'état 41 pour lequel le résultat est trivial et pour l'état 51 qui se déduit à partir de tous les autres.  
Signalons encore que la somme des éléments de cette colonne est égale au nombre moyen de jobs actifs présenté au bas.

### D. Description du tableau 3.

Ce tableau récapitulatif concerne la journée du 18/02/75.

Ce tableau est semblable au tableau 2 à ceci près que :

- les partitions 4/5/6/8/12 ont réalisés des tâches interactives.
- La partition 10 a effectué des tâches orientées "batch processing mode".



- La partition 9 a exécuté le programme de monitoring.

E. Description du tableau 4.

Ce tableau partiel concerne également la journée du 18/02/75. Ce tableau ne reprend que les partitions auxquelles étaient attribuées des travaux de saisie pure. Il s'agissait de voir si l'heure de table avait une influence significative sur les résultats globaux de la journée.

## PROGRAMME D'ECHANTILLONNAGE.

```

1. 01 S TIM=1,E="",K1=74000
1. 10 I %V(65400)/100&2.64=.64 S E(TIM-1)="FIN@"E(TIM-1) D 10 H
1. 20 S ST=%V(44),JT=%V(ST),DT=%V(ST+4)
1. 30 H 0,0,0 F I=JT,2:JT+30,JT+82,2:JT+102 S JT(I-JT/2)=%V(I)
1. 35 F I=2,46,62 S DT(I)=%V(DT+1)
1. 40 S N1=JT(0)/25600,N2=JT(0)/100
1. 50 S NJ=N1-(N2&2.55)*100
1. 51 S %M=%I(%T/60),%N=%E("AP",%I(%M/720)+1)@"M"
1. 52 S %M=-%I(%M/720)*720+%M,%H=%I(%M/60)
1. 53 S %M=-%H*60+%M I %N="PM" S %H=%H+12
1. 55 I %H?1D S %H=0&%H
1. 56 I %M?1D S %M=0&%M
1. 57 S E=E&%H@"": "%M"
1. 58 I %N?1D S %N=0&%N
1. 60 S E=E&"@"&%N@"*"
1. 90 G 2

2. 20 F I=1:1:N1*100 D 3
2. 21 S Q(45)=DT(46)/100&2.55*50 F I=2,46,62 S Q(I)=DT(I)/25600*50
2. 22 F I=2,45,46,62 D 6
2. 25 S E(TIM)=E@" ",E=""
2. 30 S TIM=TIM+1 I TIM=9 D 10
2. 40 S N=%I(%T/60)*60,N=%T-N H 60-N G 1.1

3. 10 I JT(I)/25600&1.28 D 3.3 0
3. 20 E D 3.4 I SW S X=.32 D 5
3. 30 G
3. 40 F J=41,1:51 S X=J/100 D 4 I Y=1 D 5 G
3. 50 D 4
3. 80 S X=JT(I)/25600 D 5

4. 10 S K=JT(J)/100&2.55*50,SW=1
4. 20 F D=.01:.01 W K&(DC=N1) S Y=K,K=JT(K)/100&2.55*50 I Y=1 S SW=0 G

5. 01 S X=X*100 I X=192 S X=28
5. 02 I X=196 S X=29
5. 03 I X=200 S X=30
5. 10 S E=E&X@"*"

6. 10 I Q(I)?1D S Q(I)=0&Q(I)
6. 20 S E=E&Q(I)@"*"

10. 10 A 61:0:4194304+K1 T E(1),E(2),E(3),E(4),E(5),E(6),E(7),E(8)
10. 20 S TIM=1,E="",K1=K1+1 U 61

```



[illegible]

TABLEAU 1.



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
28	/	66.33	51.25	26.63	28.39	1.50	53.51	1.25	80.15	46.98	45.72	/	/	/	51.13	4.01
29	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
30	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
32	/	2.01	1.25	3.76	3.26	3.51	2.51	6.53	0.75	6.28	5.52	/	/	/	4.50	0.35
41	/	/	/	/	/	/	/	/	/	/	/	100.0	/	/	*	*
42	/	0.75	0.75	1.25	4.02	2.51	0.50	2.26	2.51	3.01	1.75	/	/	/	2.46	0.19
43	/	4.77	1.25	3.01	14.82	8.54	2.76	16.58	3.51	10.80	13.06	/	/	/	10.07	0.79
44	/	9.29	4.02	3.76	11.55	10.84	5.77	8.04	6.53	15.82	15.57	/	/	/	11.60	0.91
45	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
46	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
47	/	0.75	0.25	0.75	1.50	2.26	0.75	1.50	0.25	1.25	2.26	/	/	/	1.47	0.11
48	/	0.50	1.75	0.50	5.77	1.75	0.75	4.52	1.75	4.02	5.77	/	/	/	3.45	0.27
49	/	3.26	2.76	1.00	13.56	13.06	0.25	24.12	3.51	11.05	10.30	/	/	/	10.55	0.82
50	/	/	/	/	0.25	/	/	35.17	1.00	0.75	/	/	/	/	4.73	0.37
51	100.0	12.31	36.68	59.29	16.83	56.03	33.16	/	/	/	/	/	100.0	100.0	*	*

ASSIGNATION PRINTER : 324 minutes

ASSIGNATION TAPE : 10 minutes

DUREE DU MONITORING : 398 minutes/ à partir de 10:19

NOMBRE MOYEN DE JOBS ACTIFS : 7.84



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
28	/	45.00	4.09	51.59	61.59	47.95	56.81	38.40	/	1.59	13.63	62.27	/	/	56.73	3.82
29	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
30	/	/	/	0.22	/	/	/	/	/	/	/	/	/	/	0.03	/
32	/	2.27	1.13	1.59	4.77	7.04	0.45	1.13	/	5.68	3.18	2.50	/	/	4.41	0.29
41	/	/	/	/	/	/	/	/	100.0	/	/	/	/	/	*	*
42	/	0.68	1.59	2.50	2.27	1.59	0.22	2.27	/	3.86	2.27	1.59	/	/	2.79	0.18
43	/	1.36	2.27	3.86	5.90	2.95	2.04	3.86	/	21.36	9.54	2.95	/	/	8.31	0.56
44	/	2.72	3.63	6.13	13.18	7.95	2.95	7.04	/	5.90	5.00	4.31	/	/	8.72	0.58
45	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
46	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
47	/	0.22	0.68	0.45	2.04	0.45	0.68	0.90	/	2.95	0.22	0.45	/	/	1.34	0.09
48	/	0.68	1.36	1.36	2.72	3.40	1.59	2.72	/	11.59	3.40	1.13	/	/	4.44	0.30
49	/	1.13	4.09	4.77	7.27	4.31	0.90	5.68	/	21.81	6.36	1.36	/	/	8.55	0.57
50	/	/	1.81	2.04	0.22	0.45	0.68	0.90	/	25.22	/	/	/	/	4.64	0.31
51	100.0	45.90	79.31	25.45	/	23.86	33.63	37.04	/	/	56.36	23.40	100.0	100.0	*	*

ASSIGNATION PRINTER : 358 minutes

NOMBRE MOYEN DE JOBS ACTIFS : 6.72

ASSIGNATION TAPE : 6 minutes

DUREE DU MONITORING : 440 minutes/ à partir de 9:32

0	1	2	3	4	5
28	54.92	56.91	56.79	63.71	46.82
29	/	/	/	/	/
30	/	/	/	/	/
32	7.04	4.47	12.96	7.07	1.58
41	/	/	/	/	/
42	1.40	3.25	2.46	1.76	2.38
43	5.63	7.31	4.32	5.30	11.90
44	16.19	14.22	9.82	15.04	15.07
45	/	/	/	/	/
46	/	/	/	/	/
47	2.81	2.43	0.61	0.88	3.17
48	2.81	3.25	6.17	4.42	4.76
49	8.45	8.13	6.17	1.76	14.28
50	0.70	/	0.61	/	/
51	/	/	/	/	/

## RECAPITULATIF DE:

- 1) Partition 5 de  
9:32 à 12:00
- 2) Partition 5 de  
13:00 à 17:09
- 3) Partition 6 de  
9:32 à 12:20
- 4) Partition 6 de  
13:02 à 14:56
- 5) Partition 8 de  
15:01 à 17:09

TABLEAU 5.



## APPENDICE. B

MUMPS : buts et historique.

MUMPS est un sigle pour " Massachusetts General Hospital Utility MultiProgramming System ". C'est aussi le nom d'un langage interprétatif de haut niveau destiné à être utilisé dans des applications de traitement de données, qui nécessitent la manipulation d'éléments de type "string" d'une banque de données. C'est également le nom du système d'exploitation destiné à traiter ce langage.

Le développement de MUMPS a commencé en 1967 au " Massachusetts General Hospital Laboratory of Computer Science " et conduisit à la production d'un article [2.] qui signifiait la naissance de MUMPS.

La viabilité technique de MUMPS fut très tôt réalisée et différentes firmes créèrent leur propre version de MUMPS, engendrant ainsi une demi-douzaine de dialectes. MUMPS-11 est de ceux-là.

En 1973 fut créé un " MUMPS Development Committee " (MDC), supporté par le Gouvernement des Etats-Unis, et dont le but principal était de réaliser des spécifications standards pour MUMPS.

La plupart des applications réalisées à l'aide des divers dialectes existants concernent le domaine de l'informatique hospitalière. (Ce domaine a donné naissance à MUMPS)

Le langage MUMPS fournit des primitives, des opérateurs, et des fonctions destinés à la manipulation souple et efficace des listes de caractères.

Cependant, la plus importante caractéristique de MUMPS est certainement sa "data base", pour sa structure et ses deux méthodes d'accès.

Tout fichier qui s'appelle dans la terminologie MUMPS variable globale, est stocké sur disque et est structuré sous forme d'arbre hiérarchisé. Toute donnée dans l'arbre est référencée par une séquence d'indices de la variable globale.

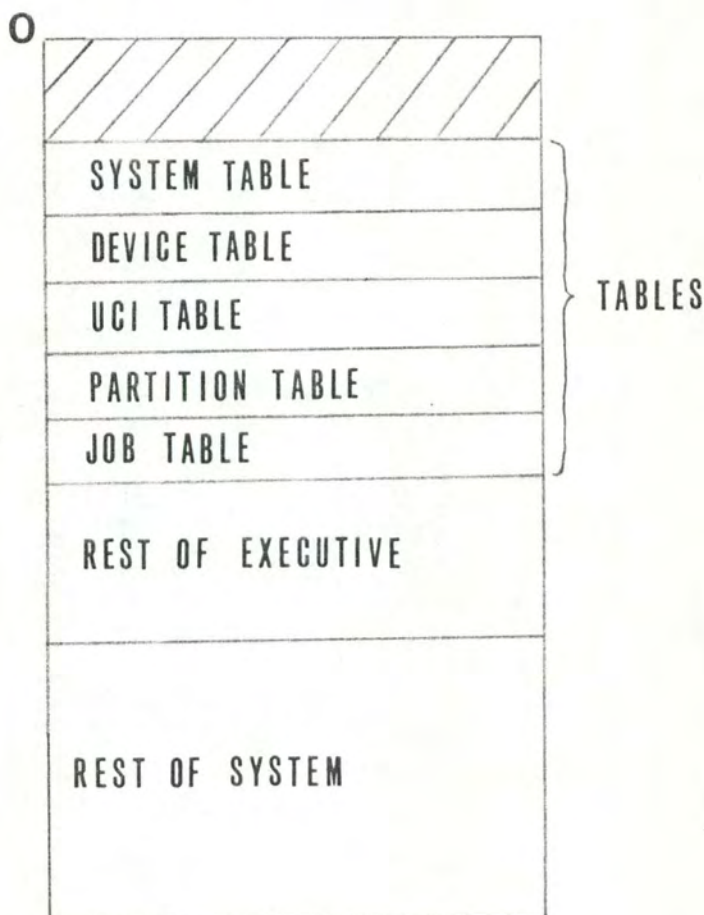
## APPENDICE.C

"MUMPS-11 SYSTEM".

Cet appendice est destiné à compléter la description de l'exécutive MUMPS-11. Une première partie concernera la description des tables du système. On présentera dans une seconde partie la description des "system queuing directives".

A. Description des tables.

Les tables du système sont physiquement contiguës en mémoire. La configuration de celle-ci est la suivante:

A.1. Description de la "System Table" (SYSTAB).

SYSTAB est le répertoire des constantes et paramètres du système.

SYSTAB contient en plus des pointeurs vers les autres tables du système. L'adresse de SYSTAB est contenue à l'adresse mémoire 4410. La structure de SYSTAB est la suivante :



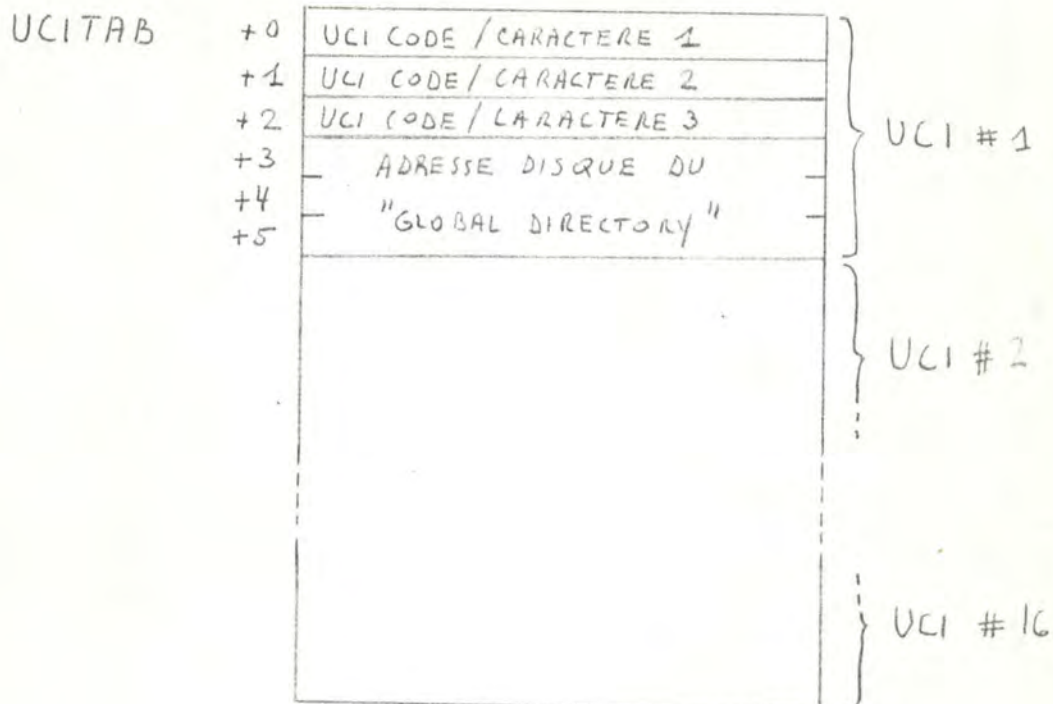
Location (decimal)	Contents
SYSTAB+0	Address of Job Table (JOBTAB)
+2	Address of Partition Table (PARTAB)
+4	Address of Device Table (DEVTAB)
+6	Base Address of Device Buffers
+8	Address of UCI Table (UCITAB)
+10	Address of Disk Storage Allocation Table
+12	Address of Garbage Table (disk blocks to be deallocated)
+14	Address of first 256-word buffer in Buffer Pool
+16	Count of number of illegal interrupts the system has detected
+18	Disk block address of data contained in UTLBUF (SYSTAB+100)
+20	Logical disk number for block in UTLBUF
+22	Address of Disk Buffer #1
+24	Address of Disk Buffer #2
+26	Garbage Indicator (0 = no garbage)
+27	Physical Disk Overflow Switch for Logical Disk. 0 (0 = overflow allowed)
+28	Number of ticks remaining in this second
+29	Number of ticks per second
+30	Number of ticks left in current job's time slice
+31	Initial value of slice (ticks) for current job
+32	Number of ticks in basic time slice
+33	Time in seconds since midnight (high-order bits)
+34	Time in seconds since midnight (low-order bits)
+36	Date in the form: (yy*500)+ddd where: yy = year -1900 ddd = day count since December 31
+38	Base address of System Stack
+40	First available address above device buffers
+42	Base address of first partition (other partitions follow sequentially)
+44	Status Register address of system clock
+46	Number of job in the run queue on system restart
+47	Number of job in the disk I/O bound queue on system restart
SYSTAB+48	Line Buffer starting address
+50	End Address of Symbol Table+1
+52	Pointer to Global Directory (3 bytes)
+55	Physical Disk Overflow Switch (0 = overflow allowed)
+56	Buffer Address given to most recent Job in Run Queue
+58	Address of System Information (3 bytes)
+61	Most recent Job in Run Queue (when 0, job is being Swapped out; error not reported until next swap-in)
+62	Programmer Access Code
+64	Initially set to CTRL/X CTRL/X XTRL/X
+66	Disk error tally counter, incremented by 1 for each disk I/O hardware error



+68	Magtape unit 0 buffer address from pool
+70	Magtape unit 1 buffer address from pool
+72	Magtape unit 2 buffer address from pool
+74	Magtape unit 3 buffer address from pool
+76	Reserved Device #0
+78	Reserved Device #1
+80	Reserved Device #2
+82	Reserved Device #3
+84	DECTape unit 0 buffer address from pool
+86	DECTape unit 1 buffer address from pool
+88	DECTape unit 2 buffer address from pool
+90	DECTape unit 3 buffer address from pool
+92	Sequential Disk Processor #0 buffer address from pool
+94	Sequential Disk Processor #1 buffer address from pool
+96	Sequential Disk Processor #2 buffer address from pool
+98	Sequential Disk Processor #3 buffer address from pool
+100	UTLBUF Address of utility buffer (view device) from pool
+102	Magtape Error Count
+104	DECTape Error Count
+106	RK11 Disk Error Count
+108	Disk Address of latest error
+112	RK11 Hardware Status Register
+114	RF11 Disk Error Count
+116	Disk Address of latest error
+120	RF11 Hardware Status Register
+122	RP11C Disk Error Count
+124	Disk Error Address: section in bits 0-3, track in bits 8-12
+126	Disk Error Address: cycle in bits 0-8, drive in bits 10-12
+128	RP11 Hardware Status Register
+130	Base Address of Ring Buffer area
+132	Base Address of Ring Buffer Queue (address of next available Ring Buffer)
SYSTAB+134	Base Address of Device Buffers for Multiplexer
+136	End Address of Device Buffers for Multiplexer
+138	RP11 Disk Drive Number Remapping Table (each byte corresponds to a physical unit (0-7; initially assigned to successive logical units 0-7 in that order)
+146	RK11 Disk Drive Number Remapping Table (each byte corresponds to a physical unit (0-7; initially assigned to successive logical units 0-7 in that order)
+154	Number of Jobs waiting to run
+155	Write-check Switch (if non-zero, all disk write operations will be checked)
+156	Index into System Bootstrap for Disk Descriptor Table
+158	Low limit (address for System Stack (LOWSTK); normally set to 420 <sub>8</sub> , but may be altered during system generation.
+160	Not used
+162	Address of Interrupt Service Routine for Multiplexer
+164	Multiplexer #1 Hardware Status Register
+166	Multiplexer #2 Hardware Status Register







#### A.4. Description de la partition table (PARTAB).

PARTAB contient les informations relatives à la "base address" et à la taille des partitions du système. Cette table débute par un mot de tête dont :

- le low byte (pair) contient la taille standard des partitions.
- le high byte contient le nombre maximum des partitions.

À la génération du système, l'utilisateur peut spécifier un maximum de 18 partitions pour des systèmes n'excédant pas 28 kmots de mémoire, ou un maximum de 40 partitions pour des systèmes de plus de 28 kmots de mémoire.

Chaque entrée de la table tient en un mot dont :

- le low byte contient la taille de la partition.
- le high byte contient les "high order bits" de la "base address" de la partition.

La taille d'une partition est spécifiée en "128 words increment" moins 1.

Par exemple, une partition de 4 kmots aurait dans le low byte de son entrée :  $4096/128 - 1 = 31_{10}$

La "base address" d'une partition est un entier multiple de 128. Par conséquent, les chiffres significatifs de la "base address" (une adresse sur un mot de 16 bits) occupent seulement les "high order bits" du mot adresse. Les digits significatifs sont stockés dans le high byte d'une entrée de "partab".

Chaque entrée de partab correspond à une entrée de JOBTAB. Le job associé à la n ième entrée de JOBTAB utilise toujours la partition spécifiée à la n ième entrée de PARTAB.



Chaque entrée de PARTAB correspond à une entrée de JOBTAB.  
Le job associé à la n ième entrée de JOBTAB utilise toujours la partition spécifiée à la n ième entrée de PARTAB.  
La valeur de n est toujours paire et appartient à l'intervalle de valeurs entières :  $[2, \text{nombre maximum de partitions} * 2]$ .

De cette façon, la valeur de  $n$  peut servir pour faire de l'accès indexé à la fois aux éléments de JOBTAB et de PARTAB.

La configuration de PARTAB est la suivante :

	HIGH BYTE	LOW BYTE
PARTAB +0	TAILLE MAXIMUM DE LA TABLE = NOMBRE MAX. DE PARTITIONS	TAILLE STANDARD DES PARTITIONS EN MULTIPLE DE 128 MOTS.
+2	"HIGH ORDER BYTE" DE LA "BASE ADDRESS"	"SIZE IN 128 WORDS INCREMENT"
+4		
⋮		

#### A.5. Description de la Job Table (JOBTAB).

A.5.1. JOBTAB est l'espace où est stocké l'information relative à toutes les files d'attente du système.

La première entrée de JOBTAB est un mot de tête qui contient les deux informations suivantes :

-low byte (pair) : nombre actuel de partitions disponible

-high byte : nombre maximum de partitions du système.

Les entrées suivantes, en nombre égal au nombre maximum de partitions du système, contiennent les informations suivantes :

-low byte (pair) : pointeur vers l'élément suivant de la file d'attente.

-high byte : informations relatives à l'état du job.







## B. Description des "SYSTEM QUEUING DIRECTIVES".

Le module de l'executive intitulé "SYSTEM DIRECTIVE" est un ensemble de routines réalisant essentiellement la manipulation des files d'attente.

Ce module fournit un "interface" entre l'exécutive et les autres modules du système d'exploitation.

En général, ce sont les autres modules du système d'exploitation (I/O monitor, globals handler, interpreter) qui déterminent l'état d'un job.

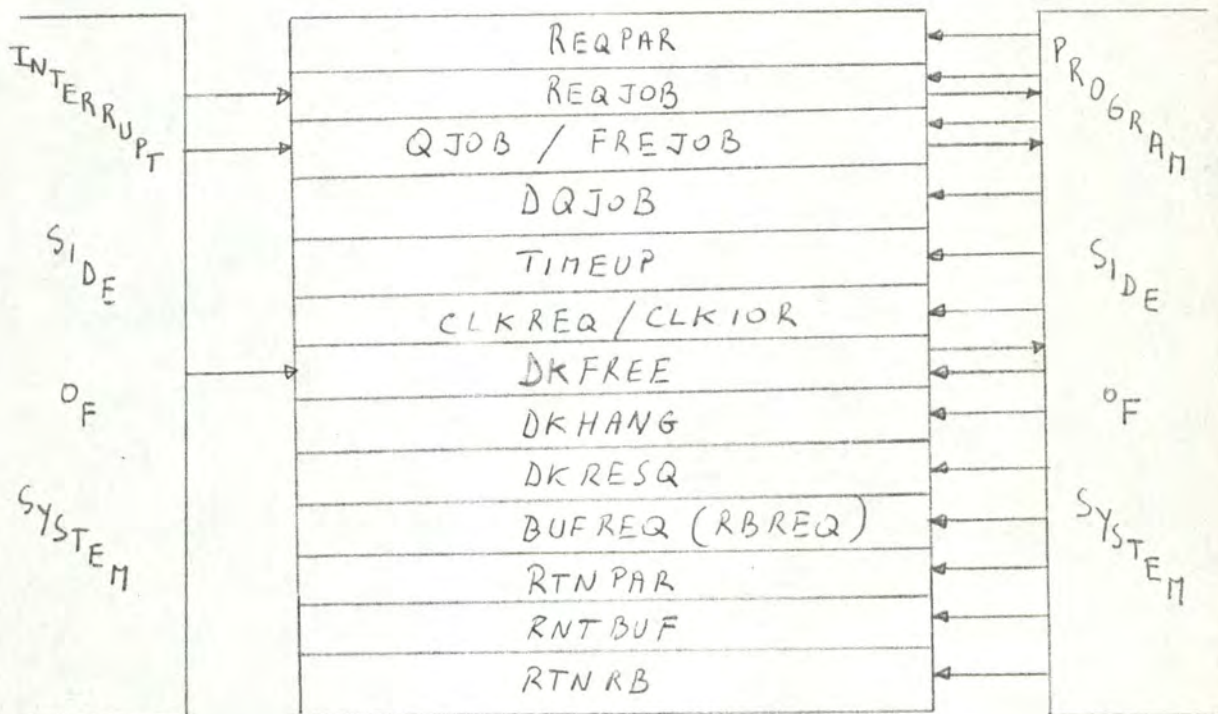
L'Executive est alors informé de ce nouvel état engendré par les autres modules, par l'intermédiaire des routines du module "System Directives".

Plus formellement, les procédures du module "System Directives" réalisent "l'interface" entre le "program side of system" et "l'interrupt side of system". (On peut définir ces deux expressions en disant que tout traitement effectué par un job lorsqu'il n'est pas dans la "Run queue" relève de "l'interrupt side of system". Tout le reste appartient au "program side of system".

Les deux règles fondamentales d'utilisation des procédures du "system directives" doivent toujours être les suivantes :

1. Le "program side of system" peut, ou bien "endormir" un job, ou bien le "réveiller".
2. L' "interrupt side of system" peut seulement "réveiller" un job.

Ces deux règles sont décrites par le schéma suivant :



Rq. : Les flèches allant vers les routines du "system directives" montrent quel "side of system" peut appeler ces routines. Les flèches partant des routines du "system directives" montrent quel "side of system" aura le contrôle quand le job recevra du traitement.



## B.1. Procédure pour démarrer un job.

### B.1.1. Directives REQJOB.

Cette directive réserve une partition d'une taille donnée ou d'une taille standard. Si une partition de la taille désirée est disponible, on place alors le "job entry number" en fin de file d'attente "Highest priority wait queue" et on le supprime de la file d'attente "Partition available queue". Si aucune partition n'est disponible, on sort de la "directive".

## B.2. Procédures de contrôle de l'état d'un job en "Run Queue".

Les 8 routines qui suivent concernent toutes le job qui est en "run queue". En conséquence, ces routines ne peuvent être appelées que du "program side of system". D'une manière générale, l'exécution de ces procédures entraînent le "swatting out" du job actuel dans la "run queue" ainsi que le transfert du contrôle à la "Idle loop".

### B.2.1. Directive DQJOB.

La procédure DQJOB oblige l'Exécutive à placer dans un état "hung" le job qui est dans la "run queue". Les divers états "hung" sont codifiés : terminal I/O hung, DECTape I/O hung, magtape I/O hung. L'état exact du job est spécifié dans le registre R5 et le job correspondant est placé dans l'entrée de JOBTAB associée au job concerné. Le contrôle est alors passé à "IDLE loop".

### B.2.2. Directive TIMEUP

La procédure TIMEUP oblige l'Exécutive à réaliser le "swap out" du job dans la "Run queue" si la tranche de temps qui lui est allouée est complètement consommée. Le job est alors placé en "wait queue" de priorité immédiatement inférieure à celle dont il est issu.

### B.2.3. Directive CLKREQ et CLKIOR.

La procédure CLKREQ oblige l'Exécutive à placer dans la file d'attente "clock queue" le job qui est en "Run queue" pour un temps spécifié par le contenu du registre R5. Si le contenu du registre est zéro, le job est simplement placé à la fin de la file d'attente "Highest priority wait queue". Dans chacun des cas, l'exécution de CLKREQ entraîne le "swapping out" du job et le transfert du contrôle à la "Idle loop".

Rq : si le nombre de ticks restant dans la seconde est inférieur à la moitié du nombre de ticks dans une seconde, alors on ajoute une unité au contenu du registre R5 qui spécifie l'intervalle de temps à passer en "clock queue".

La procédure CLKIOR est identique à CLKREQ à l'exception des 2 points suivants :



- a) l'intervalle de temps spécifié en R5 ne doit jamais être nulle.
  - b) le bit 8 du mot d'entrée de JOBTAB doit être positionné.
- Cette directive prend essentiellement en charge les instructions "READ timed".

#### B.2.4. Directive DKHANG

La directive DKHANG oblige l'Exécutive à placer le job de la "run queue" dans la file d'attente "Disk I/O bound queue". Il y a alors "swapping out" du job, et positionnement du "disk active play" de manière à indiquer que le disque est en utilisation.

#### B.2.5. Directive DKRESQ.

La directive DKRESQ est exécutée chaque fois qu'un job sollicite la ressource disque. Elle oblige l'Exécutive à placer le job de la "Run queue" à la fin de la file d'attente "Disk Resource bound queue" si le disque est utilisé. Si le disque est disponible, on sort immédiatement de la procédure.

#### B.2.6. Directive RBREQ.

La directive RBREQ est exécutée chaque fois qu'un job a besoin d'un buffer de 64 caractères (Ring buffer) et qu'aucun "Ring Buffer" n'est disponible. Elle oblige alors l'Exécutive à placer le job de la "run queue" dans la file d'attente "Ring buffer resource bound queue".

#### B.2.7. Directive BUFREQ.

La directive BUFREQ est atteinte chaque fois qu'un job a besoin d'un buffer de 256 mots et qu'il n'y a plus de buffer disponible. Elle oblige l'Exécutive à placer le job de la "run queue" dans la file d'attente "256 - W buffer resource bound queue".

#### B.2.8. Directive RTNPAR.

La directive RTNPAR caractérise la fin logique d'un job. Elle oblige l'Exécutive à placer la partition associée au job de la "run queue" dans la file d'attente des partitions disponibles.

### B.3. Procédures de contrôle de l'état d'un job qui n'est pas en "run queue".

Ces 3 directives sont utilisées pour "réveiller" des jobs qui sont, soit en "hung state", soit dans une des files d'attente "I/O or resource bound queue". Ces directives peuvent être appelées soit du "program side of system", soit de "l'interrupt side of system".



### B.3.1. Directive DKFREE.

La directive DKFREE oblige l'Exécutive à placer le job de la file d'attente "disk I/O bound queue" dans le "short time slice queue". De plus, la directive laisse la "disk I/O bound queue" dans un état "non-usable". Cette file d'attente est rendue utilisable quand le job de la "short queue" passe en "run queue". Cela permet au job qui vient d'utiliser le disque, d'avoir une haute priorité pour une nouvelle utilisation.

### B.3.2. Directive QJOB.

la directive QJOB oblige l'Exécutive à placer à la fin de la file d'attente "Highest priority wait queue", le job spécifié dans le registre R5. Le job est, soit dans un "hung state", ou dans la "clock queue". Si le registre R5 contient seulement un "job entry number", alors on cherche le job dans la "clock queue". Si le registre R5 contient un "hang code" dans son high byte (bit 15 positionné), alors on libère le job si le contenu du registre "matches" le contenu de l'entrée de "jobtab". Dans les autres cas, le contrôle est rendu au "caller" parce-que le job doit déjà être "waiting to run queue" ou suspendu pour d'autres raisons.

### B.3.3. Directive FREJOB.

La directive FREJOB oblige l'exécutive à placer à la fin de la file d'attente "Highest priority wait queue", le job spécifié dans le registre R5, si ce job est dans l'un des états ou files d'attente suivants :

- a) I/O hung.
- b) Clock queue.
- c) Disk resource bound queue.
- d) Buffer resource bound queue.
- e) Ring buffer resource bound queue.

## B.4. Directives contrôlant les "System resources" qui n'effectuent pas l'état d'un job.

### B.4.1. Directive RNTBUF.

La directive RNTBUF oblige l'Exécutive à replacer un buffer de 256-W, dont l'adresse de départ se trouve dans un registre, dans le job des buffers.

### B.4.2. Directive RTNRB.

La directive RTNRB a le même but que la directive RNTBUF, si ce n'est qu'elle s'adresse aux "ring buffers".



## B.5. Manipulation des files d'attente.

La manipulation des "system queuing directives" se fait par l'intermédiaire des quatre routines suivantes :

### B.5.1. Subroutine DQRUNQ.

La routine DQRUNQ sauve le "job's PC" et transfère le "system stack" dans le stack de l'utilisateur. Elle supprime enfin le job courant de la "run queue".

### B.5.2. Subroutine QUEUE.

La routine QUEUE a pour but d'insérer un item à la fin d'une file d'attente.

### B.5.3. Subroutine SEARCH.

La routine SEARCH recherche la file d'attente associée à un job donné. Elle initialise ensuite les registres nécessaires à la suppression de ce job dans la file d'attente trouvée.

### B.5.4. Subroutine DQUEUE.

La routine DQUEUE supprime le "job entry number" de la file d'attente dans laquelle se trouve le job considéré.



## APPENDICE. D

Description du langage MUMPS-11.A. Elément du langage.

A.1. Tous les éléments du langage sont formés à partir de l'ensemble des 128 caractères codes ASCIT ( 7 bits ).

A.2. Modes de programmation:

A.2.1. Mode direct: ce mode est utilisé pour la création, modification et stockage des programmes MUMPS-11.

A.2.2. Mode indirect: ce mode est utilisé pour l'exécution de programmes préalablement stockés.

A.3. Structure du programme:

Un programme MUMPS-11 est une séquence d'instructions que l'interpréteur MUMPS-11 traduit pour être exécutée par " l'operating system".

L'élément de base d'un programme MUMPS-11 est la ligne. Une ligne est un ensemble de 132 caractères représentant un numéro de ligne, des commandes, des arguments ou des données, suivis du caractère "line terminator".

Format général:

STEP NUMBER COMMANDE ARG<sub>1</sub>, ARG<sub>2</sub>,... COMMANDE ARG<sub>1</sub>,  
ARG<sub>2</sub>,...etc.RETURN

A.3.1. Numéro de ligne :si le numéro de ligne est absent, l'exécution est immédiate, sinon la ligne est stockée dans la partition de l'utilisateur. Le numéro de ligne identifie donc une ligne et permet d'établir la séquence d'instructions à exécuter. L'exécution d'un programme se fait séquentiellement par ordre croissant de numéro de ligne et de la gauche vers la droite dans une ligne. Un numéro de ligne est un nombre positif compris entre 0.01 et 327.67, la partie décimale ne pouvant être nulle.

A.3.2. Modules: toutes les lignes dont la partie entière du numéro de ligne est la même, forment une partie ou module, chaque module spécifiant une procédure particulière. Vis-à-vis de l'exécution, c'est le module d'un programme qui constitue l'unité de base.

A.4. Données: MUMPS-11 connaît deux types de données: soit les données numériques; soit les "string".

A.4.1. Données numériques : les nombres en MUMPS-11 sont représentés en fixe, signé, avec deux positions décimales dans l'intervalle  $\pm 21474836.47$  [ ou encore  $(2^{31} - 1)/100 \leq N \leq (2^{31} - 1)/100 ]$ .



Des résultats intermédiaires ayant plus de 2 positions décimales sont automatiquement tronqués.

A.4.2. String: c'est une séquence de 132 (ou plus) caractères légaux MUMPS-11 considérés comme une seule donnée.

#### A.5. Identificateurs :

Un identificateur est un string d'au plus 3 caractères alphanumériques, le premier étant un caractère alphabétique ou le caractère %.

#### A.6. Expression des données :

A.6.1. Littéraux : un littéral est utilisé pour spécifier un string de caractères. Le string est précédé et suivi par l'exécution du programme.

A.6.2. Constante : une constante est utilisée pour spécifier une donnée numérique qui doit appartenir à l'intervalle permis. Une constante n'est pas modifiée par une exécution du programme.

A.6.3. Variables : c'est la représentation symbolique d'une zone mémoire destinée à contenir des données qui peuvent être modifiées durant l'exécution du programme. Les variables MUMPS-11 sont de 3 types.

##### A.6.3.1. Variables locales.

Var. loc. ::= nom (ind)

Les variables locales peuvent être simples ou indicées. Les variables locales existent uniquement dans la partition du programme qui les a créés. Un indice est soit une constante; soit une variable indicée ou non; soit une expression. La valeur d'un indice doit être un nombre positif compris dans l'intervalle (0,20971.51)

A.6.3.2. Variables globales: les variables globales ont la même structure que les variables locales si ce n'est que le nom est précédé du caractère "^". Elles peuvent être simples ou indicées et dans le second cas, il n'y a pas de limite logique sur le nombre d'indices. Les variables globales sont externes à la partition du programme. Elles sont stockées sur disques selon une structure d'arbre par "l'operating system". Elles sont disponibles à tous les utilisateurs. Il existe deux types de référence d'une variable globale.

A.6.3.2.a. La référence globale : on spécifie le nom de la variable ainsi que la suite de valeur de chaque indice selon le niveau à atteindre.



A.6.3.2.b. La référence nue (Naked Reference) : par ce type de référence, on peut omettre le nom de la variable ainsi qu'une séquence (éventuellement vide) de valeurs d'indices initiaux. Cette référence nue est possible grâce à un "Naked Indicator". Ainsi, pour tout entier positif  $n$ , une référence globale à la variable  $^G(i_1, i_2, \dots, i_n)$  implique le remplissage du "Naked Indicator" par le  $n$  - uple  $(G, i_1, i_2, \dots, i_{n-1})$ . A ce moment, une référence nue telle que  $^{(s_1, s_2, \dots, s_k)}$  équivaudra à la référence globale:

$^G(i_1, i_2, \dots, i_{n-1}, s_1, s_2, \dots, s_k)$  à la suite de laquelle le "Naked Indicator" est rempli par le  $n+k-1$  uple  $(G, i_1, i_2, \dots, i_{n-1}, s_1, s_2, \dots, s_{k-1})$ .

Il est, dès lors, clair qu'une référence nue doit nécessairement succéder à une référence globale (au moins) sans quoi, le "Naked Indicator" serait indéfini.

Il est enfin à remarquer que dans une instruction du type `var.gl. = expression (var.gl., var.gl., ...)` le "Naked Indicator" est d'abord modifié par les références aux variables globales à droite du signe égal.

#### A.6.3.3. Les variables systèmes.

Ces variables qui sont "reference only" sont permanentes et mises à jour par le système. Lorsqu'elles apparaissent dans un programme, leur nom est précédé du caractère "\$". La liste des variables système MUMPS-11 est la suivante:  
 $\$A$  [DDRESS],  $\$D$  [ATE],  $\$I$  [Ø DEVICE],  
 $\$J$  [OBSTATUS],  $\$L$  [OCATION],  $\$S$  [TORAGE],  
 $\$T$  [IME],  $\$W$  [HERE],  $\$X$ ,  $\$Y$   
 (signification : voir Reference Manual)

#### B. Expressions.

Une expression est une combinaison légale d'opérandes et d'opérateurs. Les opérandes sont des littéraux, des constantes, des variables simples ou indicées, des fonctions ou des expressions enfermés dans des couples de parenthèses.

Les opérateurs sont résumés dans les 2 tableaux suivants :



a) Opération sur des données numériques.

Type	Symbole	Fonction
Arithmétique	+	addition
	-	soustraction ou opérateur unaire
	*	multiplication
	/	division
Relationnelle	<	plus petit
	>	plus grand
	=	égal
	<= ou =	plus petit ou égal
	>= ou =	plus grand ou égal
Booléenne	ou	plus petit ou plus grand
	!	ET logique
	!	OU logique
	!	NON logique

b) Opération sur des strings.

Type	Symbole	Fonction
Relationnelle		contient
		suit
Concaténation	?	"Pattern Verification"
	=	égal
		concatène

Toute expression MUMPS-11 est strictement évaluée de la gauche vers la droite. Il n'y a donc aucune règle de priorité parmi les opérateurs, excepté pour l'opérateur unaire - qui est évalué avant le booléen Not lorsque ces 2 opérateurs sont adjacents. Les conversions (String ↔ numériques) sont opérées conformément au type de l'opérateur envisagé. C'est le dernier opérateur d'une expression qui détermine le type du résultat. Une expression peut ainsi se terminer par un opérateur dont le but seul est de changer le type de résultat final.

Un (+) entraîne une conversion en numérique, un (@) entraîne la conversion en string.

C. Commandes.

Une commande est un mot-clé qui spécifie une action. Une commande est définie par sa syntaxe dans une ligne. Le premier élément d'une commande est mnémonique et caractérise la procédure à réaliser.

La définition des arguments d'une commande peut être statique ou dynamique. La définition dynamique d'un ou plusieurs arguments est réalisée par l'opérateur syntaxique d'indirection symbolisé par le caractère " ← ".

Ainsi, l'argument d'une commande peut être remplacé par le symbole ← immédiatement suivi d'un nom de variable.

A l'exécution, c'est le contenu de cette variable qui est pris comme argument. MUMPS-11 permet l'indirection uniquement pour la définition dynamique des arguments d'une commande.

Les commandes MUMPS-11 peuvent être présentées en 6 groupes fonctionnels, soit:

D.1 Commandes d'assignation.

S [ET] , K [ILL] , X [KILL] .

D.2 Commandes de contrôle.

G [OTO] , D [O] , I [F] , F [OR] , E [LSE] , C [ALL] ,  
O [VERLAY] , S [TART] , Q [UIT] , H [ANG] , H [ALT] .

D.3 Commandes I/O.

T [YPE] , R [EAD] , P [RINT] , W [RITE] , A [SSIGN] ,  
U [NASSIGN] .

D.4 Commandes "Editing".

F [ILE] , L [OAD] , E [RASE] , M [ODIFY] .

D.5 Commandes "Debugging".

B [REAK] , G [O] .

D.6 Commandes "System I/O."

V [IEW] .

(signification : voir Reference Manual).

D. Fonctions.

Les fonctions sont des procédures qui sont parties intégrantes de MUMPS-11. Il y a deux types de fonctions : les fonctions numériques qui délivrent une valeur numérique et les fonctions "string" qui délivrent un "string".

Les fonctions qui délivrent une valeur numérique peuvent être imbriquées dans d'autres fonctions, à l'opposé des fonctions qui délivrent une valeur "string".

Les fonctions MUMPS-11 peuvent être résumées par les deux tableaux suivants :



a) Type numérique.

NOM	ACTION
\$C [ REATE ]	crée un nombre unique à partir d'un string de 3 caractères.
\$D [ EFINE ]	teste le type des données.
\$F [ IND ]	recherche la position d'un caractère dans un string.
\$H [ IGH ]	fournit au niveau logique l'élément suivant d'un tableau.
\$I [ NTEGER ]	tronque une fraction en entier.
\$L [ ENGTH ]	calcule la longueur d'un string.
\$M	permet le calcul "floating point" avec des variables locales.
\$N [ EXT ]	obtenir le numéro de ligne suivant d'un tableau.
\$R [ OOT ]	recherche la racine carrée.
\$V [ IEW ]	renvoie le contenu d'une zone mémoire.

b) Type "string".

NOM	ACTION
\$A [ LTERCASE ]	conversion Majuscule/Minuscule.
\$E [ XTRACT ]	extraie les caractères contenus entre des positions spécifiées d'un string.
\$P [ IECE ]	extraie des champs spécifiques d'un string.
\$S [ TEP ]	fournit le contenu de la ligne spécifiée par son numéro.
\$T [ EXT ]	conversion de nombre en caractère ASCIT.

(spécifications : voir Reference Manual).



## APPENDICE.E

Imperfection de MUMPS-11.

Le contenu de cet appendice est constitué des défauts de MUMPS-11, tant au niveau de l'interpréteur que du système, que l'on a effectivement observé au cours des séances de programmation qui ont été nécessaires à la réalisation de ce travail.

A. Vices de fonctionnement ("BUGS") de l'interpréteur.

On a observé trois "bugs" de l'interpréteur dont un a pu être expliqué :

A.1.  $\begin{array}{l} > 1.1 \text{ } S_{\text{U}} X = "1_{\text{U}} Q" \text{ } I_{\text{U}} \leftarrow X \\ > D_{\text{U}} 1 \end{array}$

L'exécution du step 1.1. suspend complètement le système et pourtant, ce step est syntaxiquement correct.

Un step de même structure est d'ailleurs présenté en exemple dans le "Language Reference Manual" pour la commande IF, page 3 - 22.

A.2.  $\begin{array}{l} > S_{\text{U}} X = 0, Y = 3_{\text{U}} \text{ } T: X = 1_{\text{U}} X, "_{\text{U}}", Y \rightarrow \text{SYNTAX} > \phi @ \\ > S: X? 10 \text{ } X = "_{\text{U}}" @ X \rightarrow \text{SYNTAX} > \phi @ \end{array}$

L'existence d'un blanc dans les données d'un TYPE conditionnel provoque une erreur de syntaxe.

L'existence d'un blanc dans la partie droite d'une assignation (SET) conditionnelle, entraîne la production d'un message "erreur de syntaxe".

A.3.  $> 1.1 \text{ } S \text{ } N = -1 \text{ } H \text{ } N$

L'exécution du step 1.1. suspend la partition pour le reste de la journée. Cela s'explique de la façon suivante : en décrémentant l'argument négatif du HANG, on le rend encore plus négatif et on s'éloigne de ce fait de zéro, valeur pour laquelle la suspension est levée. Il n'y a pas production de message d'erreur (comme on pourrait s'y attendre, étant donné qu'il est spécifié dans le "Language Reference Manual" que l'argument du HANG doit être positif.)

Le job reste suspendu jusqu'à ce que l'argument atteigne la valeur - 65535 qui est la valeur maximum permise pour l'argument de la commande HANG.

On peut le tester en chronométrant le temps d'exécution du step 1.2. Celui-ci est de 10 secondes.

$> 1.2 \text{ } S \text{ } N = -65525 \text{ } H \text{ } N$

B. Vice de fonctionnement du système.

La séquence d'opérations décrites ci-dessous illustre le problème de la concurrence des utilisateurs, en ce qui concerne l'accès à la "data base" MUMPS-11.

a) le job A crée une variable globale à plusieurs niveaux



et désire accéder la donnée d'un certain niveau de cette variable, par exemple :  $S \uparrow TST (1,2,3) = "JOB A"$ .

- b) Le job B tue la variable globale  $\uparrow TST$ .
- c) le "Garbage Collector" collecte tout l'espace disque dont disposait la variable globale  $\uparrow TST$ .
- d) Le job C crée une nouvelle variable globale qui utilise entr'autre, le bloc disque dans lequel se trouvait la donnée  $(1,2,3)$  de la variable globale  $\uparrow TST$ . Le job C appelle cette nouvelle variable globale  $\uparrow ACD$ .
- e) Le job A poursuit son traitement par une commande SET d'assignation d'une nouvelle valeur à la donnée  $(1,2,3)$  de la variable globale  $\uparrow TST$ . Il utilise pour cela la "Naked Reference", soit :  $S \uparrow (3) = "JOB B"$ .

Le résultat de cette séquence d'opérations est qu'un bloc de la variable globale  $\uparrow ACD$  créé par l'utilisateur C a été modifié par l'utilisateur A qui croyait en fait, modifier la variable globale  $\uparrow TST$ .



# B I B L I O G R A P H I E .

= = = = =

- [1]: J.E. SHEMER, "Computer system instrumentation and performance measurement," COMPUTER, July/August 1972, vol 5, number 4.
- [2]: R. PARISEL, "Evaluation des performances d'un systeme d'exploitation," Mémoire: Facultés Universitaires Notre-Dame de la Paix/ Namur 1974.
- [3]: G. CARLSON, "Rendement d'un sous-système informatique," Communication présentée au séminaire sur la mesure de rendement, organisé en 1973 par ACM.
- [4]: H.C. LUCAS, "Performance evaluation and monitoring," COMPUTING SURVEY vol 3, September 1971.
- [5]: G.K. HUTCHINSON, "Simulation and systems evaluation," in REAL-TIME: International Computer State of the Art Report/INFOTECH.
- [6]: "Introduction to MUMPS-11 language," DEC-11-MMLTA-A-D.
- [7]: "MUMPS-11 Language Reference Manual," DEC-11-MMLMA-B-D.
- [8]: "MUMPS-11 Programmer's Guide," DEC-11-MMPGA-B-D.
- [9]: "MUMPS-11 Operator's Guide," DEC-11-MMOPA-B-D.
- [10]: J.E. SHEMER and J.B. ROBERTSON, "Instrumentation of time-shared systems," COMPUTER, July/August 1972, vol 5, number 4.
- [11]: J.M. MCKINNEY, "A survey of analytical time-sharing models," COMPUTING SURVEY, June 1969.
- [12]: N.R. NIELSON, "The simulation of time-sharing systems," Comm. of the ACM, vol 10, 397-412.
- [13]: ADL Systems, Inc., "Performance measurement : now you can find out what really goes on inside your computer," COMPUTERS and AUTOMATION, October 1973.
- [14]: F.R. ARNDT and G.M. OLIVER, "Hardware monitoring of real-time computer system performance," COMPUTER July/August 1972, vol 5, number 4.
- [15]: K.W. KOLENCE, "A software view of measurement tools," DATAMATION January 1, 1971, 32-38.
- [16]: G. CARLSON, "Utilisation des moniteurs de performances," texte présenté à l'IFIP Congress de 1972.
- [17]: T.G. PETERSON, "A comparison of software and hardware monitors," ACM/Performance Evaluation Review, June 1974, vol 3, number 2.
- [18]: "PDP 11/05/10/35/40 Processor Handbook," 1973 - DIGITAL.



- [19]: J.M. GROCHOW, "Utility functions for time-sharing system performance evaluation," COMPUTER September/October 1972 vol 5, number 5.
- [20]: R.A. GREENES, A.N. PAPPALARDO, C.W. MARBLE and G.O. BARNETT, "A system for clinical data management," AFIPS, vol 35, 1969.
- [21]: P. CALINGEART, "System performance evaluation : survey and appraisal," Comm; of the ACM, vol 10, number 1, 1967.



0. 10 ETAT DES PARTITIONS

```

1. 10 S INT=^MON(0), ST=$V(44), JT=$V(ST), DT=$V(ST+4), J1=JT+16, J2=JT+82
1. 15 S J3=JT+102, T1=ST+26, T2=ST+154, E="" A 47 P 5
1. 20 I $V(65400)/100&. 64=. 64 T E@"FIN*", ! U 47 Q
1. 30 H 0, 0, 0 S T=$T, G=$V(T1), WJ=$V(T2) F I=0, 2, 46, 62 S DT(I)=$V(DT+I)
1. 40 F I=JT: 2: J1, J2: 2: J3 S JT(I-JT/2)=$V(I)
1. 50 S N1=JT(0)/25600, N2=JT(0)/100, NJ=N1-(N2&2. 55)*100 S: NJ?1D NJ=0@NJ
1. 60 D 2 S E=E@H@": "@M@"@"@NJ@"*" F I=1: 1: N1*100 D 4
1. 70 G 3

2. 10 S M=$I(T/60), N=$E("AP", $I(M/720)+1)@"M", M=-$I(M/720)*720+M
2. 20 S H=$I(M/60), M=-H*60+M S: N="PM" H=H+12 S: H?1D H=0@H S: M?1D M=0@M

3. 05 D 5 S G=G/100&2. 55*100 I G=0 S G="00" G 3. 2
3. 10 S G="01"
3. 20 S WJ=WJ/100&2. 55*100 S: WJ?1D WJ=0@WJ T E@G@"@"@WJ@"* ", !
3. 30 S E="", N=INT-($T-T) H N G 1. 2

4. 10 G 4. 3: JT(I)/25600&1. 28 D 4. 2 S: SW E=E@"32*" Q
4. 20 F J=41: 1: 51 D 6 I Y=I S E=E@J@"*" Q
4. 30 S X=(JT(I)/25600)*100 S: X=192 X=28 S: X=196 X=29 S: X=200 X=30
4. 40 S E=E@X@"*" Q

5. 10 S Q(46)=DT(46)/100&2. 55*50 F I=0, 2, 62 S Q(I)=DT(I)/25600*50
5. 20 F I=0, 2, 46, 62 S: Q(I)?1D Q(I)=0@Q(I) S E=E@Q(I)@"*"

6. 10 S K=JT(J)/100&2. 55*50, SW=1
6. 20 F D=. 01: . 01 W K&(D<=N1) S Y=K, K=JT(K)/100&2. 55*50 I Y=I S SW=0 Q

200. 10 T "%LB" R X D "P 1024 F %LB P 2048"

```

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



0.01 ; SS SYSTEM STATUS REPORT SYSTEM UTILITY 8-26-74 V003  
 0.02 ; THIS UTILITY GENERATES A REPORT ON THE CURRENTLY ASSIGNED  
 0.03 ; DEVICE. A "SNAPSHOT" IS TAKEN OF SYSTEM TABLES AND QUEUES,  
 0.04 ; AND THE REPORT DESCRIBES THE CURRENT STATE OF EVERY JOB IN  
 0.05 ; THE SYSTEM.

0.10 ; THE REMAINDER OF PART 0 HOLDS TEXT FOR THE REPORT GENERATED.

0.19 IN RUN Q

0.20 DISK I/O DONE

0.21 DISK I/O BOUND

0.22 DISK RESOURCE BOUND

0.23 RING BUFFER RESOURCE BOUND Q

0.24 256 WORD BUFFER RESOURCE BOUND Q

0.25 HI PRIORITY WAIT Q

0.26 MED PRIORITY WAIT Q

0.27 LOW PRIORITY WAIT Q

0.28 IN CLOCK Q

0.29 AVAILABLE FOR USE

0.98 COPYRIGHT 1973 DIGITAL EQUIPMENT CORPORATION MAYNARD MASS.

1.10 K T !!!?20,"SYSTEM STATUS",!!,"DATE--" C %D T ?40,"TIME--" C %T  
 1.20 S ST=\$V(44),JT=\$V(ST),PT=\$V(ST+2),DT=\$V(ST+4),ADJ=0  
 1.25 S SQ=\$V(PT)/25600\*200+JT+2,JM=\$V(JT)/25600\*100,JA=\$V(JT)/100&2.55\*100  
 1.30 H 0,0,0 F I=JT+2:2:JM\*2+JT,SQ:2:SQ+22 S JT(I-JT/2)=\$V(I)  
 1.40 F I=DT:2:DT+94 S DT(I-DT/2)=\$V(I)  
 1.50 T !"NUMBER OF ACTIVE JOBS -- ",JM-JA  
 1.60 T ?35,"NUMBER OF AVAILABLE JOBS -- ",JA  
 1.70 F I=1:1:JM S JD(I)=""  
 1.80 S SQ=SQ-JT/2 G 2:SQ<20 S ADJ=.22 G 2

2.10 T !"JOB NUMBER",?20,"STATE",?50,"DEVICES OWNED",!  
 2.14 F N=0:1:47 F M=100,25600 S X=DT(N)/M&2.55\*50 I X,((X-JM)=<0) S JD(X)=JD(X)  
 (M/25600+(N\*2))e" "  
 2.20 F I=1:1:JM D 3  
 2.50 T !! K JT,JD,JM,DT,I,J,M,N,X,ST,SW,K,Y,D,ADJ,SQ,JA,PT

3.10 I JT(I)/25600&1.28 D 3.8  
 3.20 E D 3.4 I SW T ?5,I,?20,"IN TRANSITION"  
 3.30 I JD(I) T ?50,JD(I)  
 3.31 T ! Q  
 3.40 F J=SQ:1:SQ+10 S X=J/100 D 3.5 I T ?5,I,?20,\$S(X-ADJ) Q  
 3.50 S K=JT(J)/100&2.55\*50 S SW=1 F D=.01:.01 W K&(D<=(JM/100)) S Y=K,K=JT(K)/  
 0&2.55\*50 I Y=I S SW=0 Q  
 3.80 S X=JT(I)/25600 T ?5,I,?20,\$S(10.01+X)

11.93 TERMINAL I/O HUNG

11.97 DECTAPE I/O HUNG

12.01 MAGTAPE I/O HUNG



SYSTEM STATUS

DATE--1/8/1975

TIME--10:42

NUMBER OF ACTIVE JOBS -- 3

NUMBER OF AVAILABLE JOBS --5

JOB NUMBER

STATE

DEVICES OWNED

1	AVAILABLE FOR USE
2	AVAILABLE FOR USE
3	AVAILABLE FOR USE
4	AVAILABLE FOR USE
5	IN RUN Q
6	AVAILABLE FOR USE
7	TERMINAL I/O HUNG
8	TERMINAL I/O HUNG

3 7

4

6

INTERACTIVE SYSTEMS N.V./

BRUSSELS, BELGIUM



## 0.10 DISTRIBUTION DES OCCURRENCES DE CHAQUE ETAT.

1.10 S SP=1,D=1 R !"DEPART ? ",DEP R !"ARRIVEE ? ",ARR I '(DEP="") S SP=0,D=1  
1.20 R !"NOMBRE DE PARTITIONS : ",PAT  
1.30 F I=1:1:PAT R !"NUMERO : ",PAT(I) S PAT(I)=PAT(I)+2  
1.40 F I=28,29,30,32,41:1:51 F J=0:1:7 S T(J+(I/100))=0  
1.50 A 47 P 5 S SW=1 F H=.01:.01 W SW R X D 2  
1.55 U 47 S:ARR="" ARR=AR S K=SP-1 S: '(DEP="") K=SP  
1.60 F I=28,29,30,32,41:1:51 S ^DIS(I,0)=T(I/100) F J=1:1:7 S ^(J)=T(J+(I/100))  
1.70 S ^EFF(0)=K,^(1)=DEP,^(2)=ARR,^(3)=PAT F J=1:1:PAT S ^(J+3)=PAT(J)-2  
  
2.10 I \$P(X,"\*",1)="FIN"!(\$P(X,"\*",1)=ARR) S SW=0 Q  
2.20 Q: SP=0&('(\$P(X,"\*",1)=DEP)) S SP=SP+1,AR=\$P(X,"\*",1)  
2.25 I D S D=0,DEP=AR  
2.30 F I=28,29,30,32,41:1:51 S W(I)=0  
2.40 F L=1:1:PAT S I=\$P(X,"\*",PAT(L)),W(I)=W(I)+1  
2.50 F I=28,29,30,32,41:1:51 S N=W(I)+(I/100),T(N)=T(N)+1  
  
200.10 T "LBO" R X D "F LBO"

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



LB1 1/8/1975 10. 68

LB1

0. 01 HISTOGRAMME

```
1. 10 A 3 T # S INC=^EFF(0)/60, PG=0
1. 20 S K=$H(^DIS(-1)) F M=0:0:1 D 1. 25, 1. 3, 1. 4, 1. 5 S K=$H(^DIS(K)) G 1. 5:K<0
1. 25 I PG>4 T # S PG=0
1. 30 S PG=PG+1 T !!!?30, " FILE D'ATTENTE ", K, !!
1. 40 S J=$H(^K, -1)) F L=0:0:1 D 2 S J=$H(^J)) Q: J<0
1. 50 ;

2. 10 S X=^(J) T: X>0 J, ?5, X, ?10 F I=1: INC: X T "*"
2. 20 T: $X>0 !

200. 10 T "LB1" R X D "F LB1"
```

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



LB2 1/8/1975

10. 68

LB2

0. 10 LISTING DU MONITORING

1. 10 A 3 T #!, " TIME\*NA\* 1\* 2\* 3\* 4\* 5\* 6\* 7\* 8\*CN\*PR\*VI\*VI\*GB\*WJ\*", ! F K=1:  
T "\*"   
1. 20 T !! A 47 P 5   
1. 30 R X G 1. 4: \$P(X, "\*", 1)="FIN" D 1. 5   
1. 40 U 47, 3 Q   
1. 50 A 3 T \$P(X, "\*", 1, 16)@"\*", ! A 47 G 1. 3   
  
200. 10 T "LB2" R X D "F LB2"

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



## 0.10 LISTING % DU MONITORING

```

1. 10 S SP=1,D=1,INT=^MON(0),JOB=0,SW=1,B=" " R !"DEPART ? ",DEP
1. 20 S ST=DEP R !"ARRIVEE ? ",ARR I '(DEP="") S SP=0,D=0
1. 30 R !"NOMBRE DE PARTITIONS : ",PAT
1. 40 F I=1:1:PAT R !"NUMERO : ",PAT(I) S PAT(I)=PAT(I)+2
1. 50 F I=11,12,15,16,28,29,30,32,41:1:51 S T(I)=0
1. 60 A 47 P 5 F N=0:1 W SW R X D 2
1. 70 U 47 A 3 T #,! T "*** " C %D T ?28, "MOYENNE",?46, "POURCENTAGE"
1. 80 F I=28,29,30,32,41:1:50 D 4
1. 90 G 3

2. 10 I $P(X,"*",1)="FIN"!($P(X,"*",1)=ARR) S SW=0 Q
2. 20 Q: SP=0&('($P(X,"*",1)=DEP)) S SP=SP+1 I D S D=0,ST=$P(X,"*",1)
2. 30 F L=1:1:PAT S A=$P(X,"*",PAT(L)),T(A)=T(A)+1
2. 40 F L=11,12,15 S A=$P(X,"*",L) I '(A=0) S T(L)=T(L)+1
2. 50 S A=$P(X,"*",16),T(16)=T(16)+A,J1=$P(X,"*",2),JOB=JOB+J1

3. 10 T !"* ASSIGNATION CONSOLE = ",T(11)*INT," SECONDES. "
3. 20 T !"* ASSIGNATION PRINTER = ",T(12)*INT," SECONDES. "
3. 30 T !"* GARBAGE COLLECTOR = ",T(15)*INT," SECONDES. "
3. 40 T !"* NOMBRE DE JOBS ACTIFS = ",JOB/K-1
3. 50 T !"* JOBS WAITING TO RUN = ",T(16)/K
3. 60 T !"* HEURE DE DEPART = ",ST
3. 70 T !"* DUREE DU MONITORING = ",K*INT," SECONDES. "
3. 80 T !"* INTERVALLE DE MONITORING = ",INT," SECONDES. "
3. 90 T !!"* PARTITION NUMERO : / " F I=1:1:PAT T PAT(I)-2," / "
3. 99 T ! U 3 Q

4. 10 S K=SP-1 S: '(DEP="") K=SP T !$(4+(I/100)) S Z=T(I)/K D 5 T ?30,Z
4. 20 S Z=T(I)*100/(K*PAT-T(51)) D 5 T ?50,Z@" %" Q
4. 28 "HUNG TERMINAL" STATE
4. 29 "HUNG DECTAPE" STATE
4. 30 "HUNG MAGTAPE" STATE
4. 32 TRANSITION STATE
4. 41 RUN QUEUE
4. 42 SHORT QUEUE
4. 43 DISK I/O BOUND QUEUE
4. 44 DISK RESSOURCE BOUND QUEUE
4. 45 RING BUFFER QUEUE
4. 46 256 W BUFFER QUEUE
4. 47 HIGH WAIT QUEUE
4. 48 MEDIUM WAIT QUEUE
4. 49 LOW WAIT QUEUE
4. 50 CLOCK QUEUE

5. 10 S Z=Z@" S: $F(Z,".")=0 Z=Z@" 00" S Z1=$P(Z,".",1) S: $L(Z1)<2 Z1=B@Z1
5. 20 S Z2=$P(Z,".",2) S: $L(Z2)<2 Z2=Z2@0 S Z=Z1@". "@Z2 I Z=" 0.00" S Z=" *

200.10 T "LB3" R X D "F LB3"

```



## 0. 10 LISTING PAR PARTITION DU MONITORING

```

1. 10 R !"DEPART ? ", DEP R !"ARRIVEE ? ", ARR R !"PARTITION ? ", PAT
1. 20 S B=" ", SS=3, SP=1, INT=^MON(0) S: '(DEP="") SP=0 S: '(PAT="") SS=PAT+2
1. 30 F J=11, 13, 28, 29, 30, 32, 41: 1: 51 S T(J)=0
1. 40 A 47 P 5 S SW=1 F N=0: 1 W SW R X D 3
1. 50 U 47 D 2 G 1. 6: '(PAT="") S SS=SS+1 S: '(DEP="") SP=0
1. 55 G 1. 3: '(SS=11)&(PAT="")
1. 60 R !"CONTINUE ? ", C G 1. 1: C="Y" Q

```

```

2. 10 S K=N-1 S: '(DEP="") K=SP A 3 T #, !, ?46, "POURCENTAGE"
2. 20 F I=28, 29, 30, 32, 41: 1: 51 D 2. 6: 2. 7, 2. 8, 2. 9
2. 30 T !!"* DUREE DU MONITORING = ", K*INT, " SECONDES. "
2. 35 T !!"* INTERVALLE DE MONITORING = ", INT, " SECONDES. "
2. 40 T !!"* NUMERO DE LA PARTITION = ", SS-2, !
2. 50 T: '(DEP="") "DE"@B, DEP T: '(ARR="") B@"A"@B, ARR T ! U 3 Q
2. 60 T !$(4+(I/100)) S Z=T(I)*100/K@ S: $F(Z, ". ")=0 Z=Z@". 00"
2. 70 S Z1=$P(Z, ". ", 1) S: $L(Z1)<2 Z1=B@Z1 S Z2=$P(Z, ". ", 2)
2. 80 S: $L(Z2)<2 Z2=Z2@0 S Z=Z1@". "@Z2 I Z=" 0. 00" S Z=" * "
2. 90 T ?50, Z@" %"

```

```

3. 10 I $P(X, "*", 1)="FIN"!($P(X, "*", 1)=ARR) S SW=0 Q
3. 20 Q: SP=0&('($P(X, "*", 1)=DEP)) S SP=SP+1, A=$P(X, "*", SS)
3. 30 S T(A)=T(A)+1 F L=11, 13 S A=$P(X, "*", L) S: '(A=0) T(L)=T(L)+1

```

```

4. 28 "HUNG TERMINAL" STATE
4. 29 "HUNG DECTAPE" STATE
4. 30 "HUNG MAGTAPE" STATE
4. 32 TRANSITION STATE
4. 41 RUN QUEUE
4. 42 SHORT QUEUE
4. 43 DISK I/O BOUND QUEUE
4. 44 DISK RESSOURCE BOUND QUEUE
4. 45 RING BUFFER QUEUE
4. 46 256 W BUFFER QUEUE
4. 47 HIGH WAIT QUEUE
4. 48 MEDIUM WAIT QUEUE
4. 49 LOW WAIT QUEUE
4. 50 CLOCK QUEUE
4. 51 PARTITION AVAILABLE QUEUE

```

```

200. 10 T "LB4" R X D "F LB4"

```

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



%B0

24/7/1975 12.83

%B0

0.10 MESURE DU TEMPS D'EXECUTION.

0.20 PROCEDURE D'EXTRACTION.

1.10 S  $TTC(6)=1000$  F  $N=1:1:5$  S  $TTC(N)=N*100$

1.20 S  $ST=\$V(44)$ ,  $JT=\$V(ST)$ ,  $DT=\$V(ST+4)$ ,  $J1=JT+16$

1.30 S  $J2=JT+80$ ,  $J3=JT+102$ ,  $T3=ST+26$ ,  $T4=ST+154$

1.40 F  $J=1:1:6$  D 2 A 3 T ?10,  $TTC(J)$ , ?25, T, ?50, TTR, !

1.50 U 3 Q

2.10 S  $T=\$T$  F  $I=1:1:TTC(J)$  D 3

2.20 S  $T=\$T-T$ ,  $TTR=T*1000/TTC(J)$

3.10 S  $P=\$T$ ,  $G=\$V(T3)$ ,  $WJ=\$V(T4)$  F  $K=0, 2, 46, 62$  S  $DT(K)=\$V(DT+K)$

3.20 F  $K=JT:2:J1$ ,  $J2:2:J3$  S  $JT(K-JT/2)=\$V(K)$

200.10 T "%B0" R X D "P 1024 F %B0 P 2048"

INTERACTIVE SYSTEMS N.V./

BRUSSELS, BELGIUM



%MM

24/7/1975 12.81

%MM

0.10 MESURE DU TEMPS D'EXECUTION.  
0.20 PROCEDURE D'EXTRACTION.

1.01 S EXP=1  
1.05 S H=2.1+EXP A 3 T #, !!!, \$S(H), !!!!! U 3  
1.20 S ST=\$V(44), JT=\$V(ST), DT=\$V(ST+4), J1=JT+16  
1.30 S J2=JT+80, J3=JT+102, T3=ST+26, T4=ST+154  
1.40 A O R !"COUNT : ", TTC G 1.5: TTC="" D 2 A 3 T ?10, TTC, ?25, T, ?50, TTR, ! G  
1.50 S EXP=EXP+1 I EXP>7 U 3 Q  
1.60 G 1.05  
  
2.10 S T=\$T F I=1:1: TTC D 2+EXP  
2.20 S T=\$T-T, TTR=T\*1000/TTC  
  
3.10 S P=\$T, G=\$V(T3), WJ=\$V(T4)  
  
4.10 F K=0, 2, 46, 62 S DT(K)=\$V(DT+K)  
  
5.10 F K=JT: 2: J1, J2: 2: J3 S JT(K-JT/2)=\$V(K)  
  
6.10 F K=0, 2, 46, 62 ;  
  
7.10 S DT(K)=\$V(DT+K)  
  
8.10 F K=JT: 2: J1, J2: 2: J3 ;  
  
9.10 S JT(K-JT/2)=\$V(K)  
  
200.10 T "%MM" R X D "P 1024 F %MM P 2048"

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



MEG 1/8/1975 10. 70

MEG

1. 10 A O R ! "COUNT : ", TTC Q: TTC=""

1. 20 S T=\$T F I=1: 1: TTC D 1. 99

1. 30 S T=\$T-T, TTR=T\*1000/TTC A 3 T ?10, TTC, ?25, T, ?50, TTR, ! G 1. 1

1. 99 ; INSTRUCTION A MESURER.

200. 10 T "MEG" R X D "F MEG"

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



%L1 1/8/1975 10. 76

%L1

0. 10 TIMING TEST ROUTINE.

1. 10 R !"LINE: ", TTL S TTS="1 "@TTL  
1. 20 R !"COUNT: ", TTC G 1: TTC=0 D 2  
1. 30 S TTT=\$T F I=1:1: TTC I ←TTS  
1. 40 S TTT=\$T-TTT D 3 G 1.2  
1. 50 S ^{(0)}=200 Q

2. 10 R !"VARS: ", TTV Q: TTV="" S ←TTV

3. 10 A 3 T !, "TIMING TESTS. "  
3. 20 T !!"COUNT SECONDS EXECUTED STATEMENT", !  
3. 25 S TTR=TTT\*1000/TTC  
3. 30 T !?36, "VARS: ", TTV, !, TTC, ?11-\$L(TTT), TTT, ?18, TTR, ?28, TTL, ! U 3

200. 10 T "%L1" R X D "P 1024 F %L1 P 2048"

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



%E1

24/7/1975 12.81

%E1

0.10 MESURE DE TEMPS D'EXECUTION.

1.05 A O R ! "COUNT: ", TTC Q: TTC=""

1.07 S ST=\$V(44), T1=ST+28, T2=ST+30, T3=ST+31

1.10 S T=\$T F I=1:1: TTC D 2

1.20 S T=\$T-T, TTR=T\*1000/TTC A 3 T ?10, TTC, ?25, T, ?50, TTR, ! G 1.05

2.10 S TH=\$T, A=\$V(T1)/100&2.55\*100, B=\$V(T2)/100&2.55\*100, C=\$V(T3)/25600\*100

200.10 T "%E1" R X D "P 1024 F %E1 P 2048"

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



0.10 ETAT DES PARTITIONS

0.20 MESURE DU TEMPS D'EXECUTION.

```

1.10 S INT=^MON(0),E="",ST=$V(44),JT=$V(ST),DT=$V(ST+4)
1.15 S T1=ST+28,T2=ST+30,T3=ST+31,L=0
1.20 Q:L>60 I $V(65400)/100&2.64=.64 T E@"FIN*",! U 47 Q
1.25 S L=L+1 D 10
1.30 H 0,0,0 S L=L+1 D 10 S T=$T F I=0,2,46,62 S DT(I)=$V(DT+I)
1.35 S L=L+1 D 10
1.40 F I=JT:2:JT+16,JT+82:2:JT+102 S JT(I-JT/2)=$V(I)
1.45 S G=$V(ST+26)/100&2.55*100,WJ=$V(ST+154)/100&2.55*100,L=L+1 D 10
1.50 S N1=JT(0)/25600,N2=JT(0)/100,NJ=N1-(N2&2.55)*100 S:NJ?1D NJ=0@NJ
1.60 D 2 S E=E@H@": "@M@""@NJ@"" F I=1:1:N1*100 D 4
1.70 G 3

2.10 S M=$I(T/60),N=$E("AP",$I(M/720)+1)@"M",M=-$I(M/720)*720+M
2.20 S H=$I(M/60),M=-H*60+M S:N="PM" H=H+12 S:H?1D H=0@H S:M?1D M=0@M

3.05 D 5 I G=0 S G="00" G 3.2
3.10 S G="01"
3.20 S:WJ?1D WJ=0@WJ S E=E@G@""@WJ@""
3.30 T E@" ",! S E="",N=INT-($T-T),L=L+1 D 10 H.N S L=L+1 D 10 G 1.2


4.10 G 4.3:JT(I)/25600&1.28 D 4.2 S:SW E=E@"32*" Q
4.20 F J=41:1:51 D 6 I Y=I S E=E@J@"" Q
4.30 S X=(JT(I)/25600)*100 S:X=192 X=28 S:X=196 X=29 S:X=200 X=30
4.40 S E=E@X@"" Q

5.10 S Q(46)=DT(46)/100&2.55*50 F I=0,2,46,62 S Q(I)=DT(I)/25600*50
5.20 F I=0,2,46,62 S:Q(I)?1D Q(I)=0@Q(I) S E=E@Q(I)@"*"

6.10 S K=JT(J)/100&2.55*50,SW=1
6.20 F D=.01:.01 W K&(D<=N1) S Y=K,K=JT(K)/100&2.55*50 I Y=I S SW=0 Q

10.10 S T(L)=$T,A(L)=$V(T1)/100&2.55*100
10.20 S B(L)=$V(T2)/100&2.55*100,C(L)=$V(T2)/25600*100
200.10 T "XE2" R X D "P 1024 F XE2 P 2048"

```


 PRELEVEMENT  
DU TEMPS.

 INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



XE3

24/7/1975 12.82

XE3

0.10 ETAT DES PARTITIONS : EXTRACTEUR.  
0.20 MESURE DU TEMPS D'EXECUTION.

```

1.10 S INT=^MON(0),KA=0,ST=$V(44),JT=$V(ST),DT=$V(ST+4)
1.15 S T1=ST+28,T2=ST+30,T3=ST+31,L=0
1.20 S A="",B="" Q:L>60 S T=$T I $V(65400)/100&.64=.64 D ^DU(0)=KA-1 Q
1.25 H 0,0,0 S L=L+1 D 5
1.30 F I=JT:2:JT+16 S A=A@$V(I)@"*"
1.32 S L=L+1 D 5
1.35 F I=JT+82:2:JT+102 S B=B@$V(I)@"*"
1.37 S L=L+1 D 5
1.40 F I=0,2,46,62 S A=A@$V(DT+I)@"*"
1.50 S A=A@T@"*"@$V(ST+26)@"*"@$V(ST+154)@"*"
1.55 S L=L+1 D 5
1.60 S ^MON(KA)=A,^(KA+1)=B,KA=KA+2,L=L+1 D 5
1.70 H INT-($T-T) S L=L+1 D 5 G 1.2

5.10 S T(L)=$T,A(L)=$V(T1)/100&2.55*100
5.20 S B(L)=$V(T2)/100&2.55*100,C(L)=$V(T3)/25600*100

200.10 T "XE3" R X D "P 1024 F XE3 P 2048"

```

← PRELEVEMENT  
DU TEMPS.

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



0.10 ETAT DES PARTITIONS : EXTRACTEUR.  
0.20 MESURE DU TEMPS D'EXECUTION.

1.10 S INT=^MON(0),KA=1,ST=\$V(44),JT=\$V(ST),DT=\$V(ST+4)  
1.12 S J1=JT+30,J2=JT+82,J3=JT+102,T3=ST+26,T4=ST+154  
1.15 S T1=ST+28,T2=ST+30,L=0  
1.20 Q:L>30 S A=" ",B=" " I \$V(65400)/100&.64=.64 S ^(.5)=KA-1 Q  
1.25 H 0,0,0 S L=L+1 D 5  
1.30 S T=\$T,G=\$V(T3),WJ=\$V(T4) F I=0,2,46,62 S DT(I)=\$V(DT+I)  
1.40 F I=JT:2:J1,J2:2:J3 S JT(I-JT/2)=\$V(I)  
1.45 S L=L+1 D 5  
1.50 F I=1:1:15 S A=A@JT(I)@"\*"  
1.60 F I=0,2,46,62 S A=A@DT(I)@"\*"  
1.70 S A=A@T@"\*"@G@"\*"@WJ@"\*" F I=41:1:51 S B=B@JT(I)@"\*"  
1.80 S ^ (KA)=A,^(KA+1)=B,KA=KA+2,L=L+1 D 5 H INT-(\$T-T) G 1.2

5.10 S T(L)=\$T,A(L)=\$V(T1)/100&2.55\*100  
5.20 S B(L)=\$V(T2)/100&2.55\*100,C(L)=\$V(T2)/25600\*100

PRELEVEMENT  
DU TEMPS.

200.10 T "%E4" R X D "P 1024 F %E4 P 2048"

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



0. 10 MANIPULATION DU PROCESSOR STATUS WORD.

```

1. 10 S ST=$V(44), T28=ST+28, T30=ST+30, J=0 F I=1:1:5 D 2
1. 20 D 3 A 46 P 256 V 65534: $V(65534)+192 U 46 D 3 F I=6:1:15 D 2
1. 30 D 3 A 46 P 256 V 65534: $V(65534)/100&653. 11*100 U 46 D 3
1. 40 F I=16:1 20 D 2
1. 50 D 4 Q

2. 10 S T(I)=$T, A(I)=$V(T28)/100&2. 55*100
2. 20 S B(I)=$V(T30)?100&2. 55*100, C(I)=$V(T30)/25600*100

3. 10 S J=J+1, PSW(J)=$V(65534), PRT(J)=PSW(J)/100&2. 24*100/32

4. 05 A 3 T # S K=1, IN=1, OUT=5
4. 10 T !!, $S(4. 9), ! F I=IN:1:OUT T ?5, T(I), ?20, A(I), ?30, B(I), ?40, C(I), !
4. 20 T !!, $S(4. 91), !, ?5, "PSW= ", PSW(K), ?20, "PRIORITE= ", PRT(K), !
4. 30 T !!, $S(4. 92), !! S K=2, IN=6, OUT=15 D 4. 2, 4. 1
4. 40 S K=3 D 4. 2 T !!, $S(4. 92), !! S K=4, IN=16, OUT=20 D 4. 2, 4. 1 U 3
4. 90 ***** OBSERVATION DU TEMPS. *****
4. 91 ***** OBSERVATION DU PROCESSOR STATUS WORD. *****
4. 92 ***** MODIFICATION DU PROCESSOR STATUS WORD. *****

200. 10 T "%B2" R X D "P 1024 F %B2 P 2048"
    
```

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



0.10 ETAT DES PARTITIONS : EXTRACTEUR.

```

1.10 S INT=^MON(0),ST=$V(44),JT=$V(ST),DT=$V(ST+4),J1=JT+18
1.15 S J2=JT+82,J3=JT+102,T3=ST+26,T4=ST+154,KA=1
1.20 S A="",B="" I $V(65400)/100&.64=.64 S ^(.5)=KA-1 Q
1.30 H 0,0,0 S T=$T,G=$V(T3),WJ=$V(T4) F I=0,2,46,62 S DT(I)=$V(DT+I)
1.40 F I=JT:2:J1,J2:2:J3 S JT(I-JT/2)=$V(I)
1.50 F I=0:1:9 S A=A@JT(I)@"*"
1.60 F I=0,2,46,62 S A=A@DT(I)@"*"
1.70 S A=A@T@"*"@G@"*"@WJ@"*" F I=41:1:51 S B=B@JT(I)@"*"
1.80 S ^(KA)=A,^(KA+1)=B,KA=KA+2 H INT-($T-T) G 1.2

```

200.10 T "%L2" R X D "P 1024 F %L2 P 2048"

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



# 0.01 ETAT DES PARTITIONS : ANALYSEUR.

```

1.05 S LIM=^MON(0.5),LK=1,E="" A 47 P 5
1.10 I LK>LIM T E@"FIN*",! U 47 Q
1.15 S A=^MON(LK),B=^MON(LK+1) F I=1:1:10 S JT(I-1)=$P(A,"",I)
1.20 F I=1:1:11 S JT(I+40)=$P(B,"",I)
1.25 S DT(0)=$P(A,"",11),DT(2)=$P(A,"",12),DT(46)=$P(A,"",13)
1.30 S DT(62)=$P(A,"",14),T=$P(A,"",15)
1.50 S N1=JT(0)/25600,N2=JT(0)/100,NJ=N1-(N2&2.55)*100 S:NJ?1D NJ=0@NJ
1.60 D 2 S E=E@H@": "@M@"@"@NJ@"@" F I=1:1:N1*100 D 4
1.70 G 3

2.10 S M=$I(T/60),N=$E("AP",$I(M/720)+1)@"M",M=-$I(M/720)*720+M
2.20 S H=$I(M/60),M=-H*60+M S:N="PM" H=H+12 S:H?1D H=0@H S:M?1D M=0@M

3.05 D 5 S G=$P(A,"",16)/100&2.55*100 I G=0 S G="00" G 3.2
3.10 S G="01"
3.20 S WJ=$P(A,"",17)/100&2.55*100 S:WJ?1D WJ=0@WJ S E=E@G@"@"@WJ@"@"
3.30 T E@" ",! S E="",LK=LK+2 G 1.1

4.10 G 4.3:JT(I)/25600&1.28 D 4.2 S:SW E=E@"32*" Q
4.20 F J=41:1:51 D 6 I Y=I S E=E@J@"*" Q
4.30 S X=(JT(I)/25600)*100 S:X=192 X=28 S:X=196 X=29 S:X=200 X=30
4.40 S E=E@X@"*" Q

5.10 S Q(46)=DT(46)/100&2.55*50 F I=0,2,62 S Q(I)=DT(I)/25600*50
5.20 F I=0,2,46,62 S:Q(I)?1D Q(I)=0@Q(I) S E=E@Q(I)@"*"

6.10 S K=JT(J)/100&2.55*50,SW=1
6.20 F D=.01:.01 W K&(D<=N1) S Y=K,K=JT(K)/100&2.55*50 I Y=I S SW=0 Q

200.10 T "%L3" R X D "P 1024 F %L3 P 2048"

```

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



0.10 ETAT DES PARTITIONS : EXTRACTEUR.  
0.20 MASQUE PROCESSOR STATUS WORD.

1.10 S INT= $\wedge$ MON(0), ST=\$V(44), JT=\$V(ST), DT=\$V(ST+4), J1=JT+18  
1.15 S J2=JT+82, J3=JT+102, T3=ST+26, T4=ST+154, KA=1  
1.20 S A="", B="" I \$V(65400)/100&.64=.64 S  $\wedge$ (0.5)=KA-1 Q  
1.25 A 46 P 256 V 65534: \$V(65534)+192 U 46  
1.30 S T=\$T, G=\$V(T3), WJ=\$V(T4) F I=0,2,46,62 S DT(I)=\$V(DT+I)  
1.40 F I=JT:2:J1, J2:2:J3 S JT(I-JT/2)=\$V(I)  
1.45 A 46 P 256 V 65534: \$V(65534)/100&.653.11\*100 U 46  
1.50 F I=0:1:9 S A=A@JT(I)@"\*"  
1.60 F I=0,2,46,62 S A=A@DT(I)@"\*"  
1.70 S A=A@T@"\*"@G@"\*"@WJ@"\*" F I=41:1:51 S B=B@JT(I)@"\*"  
1.80 S  $\wedge$ (KA)=A,  $\wedge$ (KA+1)=B, KA=KA+2 H INT-(\$T-T) G 1.2

200.10 T "%MB" R X D "P 1024 F %MB P 2048"

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



0.10 ETAT DES PARTITIONS

0.20 MESURE DU TEMPS D'EXECUTION.

1.10 S INT= $\wedge$ MON(0), L=0, ST=\$V(44), JT=\$V(ST), DT=\$V(ST+4), J1=JT+30, J2=JT+82  
 1.15 S J3=JT+102, T1=ST+28, T2=ST+30, T3=ST+26, T4=ST+154, E="", CHG=1 A 47 P 5  
 1.20 G 7: L>31 I \$V(65400)/100&.64=.64 T E@FIN\*, ! U 47 Q  
 1.25 H 0,0,0 S L=L+1 D 10  
 1.30 S T=\$T, G=\$V(T3), WJ=\$V(T4) F I=0,2,46,62 S DT(I)=\$V(DT+I)  
 1.40 F I=JT:2:J1, J2:2:J3 S JT(I-JT/2)=\$V(I)  
 1.45 S L=L+1 D 10  
 1.50 S N1=JT(0)/25600, N2=JT(0)/100, NJ=N1-(N2&2.55)\*100 S: NJ?1D NJ=0@NJ  
 1.60 D 2 S E=E@H@": "E@E"\*@NJ@E"\* F I=1:1:N1\*100 D 4  
 1.70 G 3

2.10 S M=\$I(T/60), N=\$E("AP", \$I(M/720)+1)@"M", M=-\$I(M/720)\*720+M  
 2.20 S H=\$I(M/60), M=-H\*60+M S: N="PM" H=H+12 S: H?1D H=0@H S: M?1D M=0@M

3.05 D 5 S G=G/100&2.55\*100 I G=0 S G="00" G 3.2  
 3.10 S G="01"  
 3.20 S WJ=WJ/100&2.55\*100 S: WJ?1D WJ=0@WJ S E=E@G@E"\*@WJ@E"\*  
 3.30 T E@ " , ! S E="", N=INT-(\$T-T), L=L+1 S: N<0 N=1 D 10 H N G 1.2

4.10 G 4.3: JT(I)/25600&1.28 D 4.2 S: SW E=E@32\* Q  
 4.20 F J=41:1:51 D 6 I Y=I S E=E@J@E"\* Q  
 4.30 S X=(JT(I)/25600)\*100 S: X=192 X=28 S: X=196 X=29 S: X=200 X=30  
 4.40 S E=E@X@E"\* Q

5.10 S Q(46)=DT(46)/100&2.55\*50 F I=0,2,46,62 S Q(I)=DT(I)/25600\*50  
 5.20 F I=0,2,46,62 S: Q(I)?1D Q(I)=0@Q(I) S E=E@Q(I)@E"\*

6.10 S K=JT(J)/100&2.55\*50, SW=1  
 6.20 F D=.01:.01 W K&(D<=N1) S Y=K, K=JT(K)/100&2.55\*50 I Y=I S SW=0 Q

7.10 A 3 T #, !!, "CHARGE DU SYSTEME : ", CHG, !!!  
 7.20 T ?8, "\$T", ?19, "S28", ?31, "S30", ?43, "S31", !!  
 7.30 F L=1:1:30 T ?5, T(L), ?20, A(L), ?32, B(L), ?43, C(L), !  
 7.50 I CHG<7 S %L9 D 7.7 S CHG=CHG+1, L=0 G 1.25 ← CHARGE DU SYSTEME  
 7.60 U 3 Q  
 7.70 F M=1:1:30 K T(M), A(M), B(M), C(M)

10.10 S T(L)=\$T, A(L)=\$V(T1)/100&2.55\*100  
 10.20 S B(L)=\$V(T2)/100&2.55\*100, C(L)=\$V(T2)/25600\*100

← PRELEVEMENT  
DU TEMPS.

200.10 T "%L7" R X D "P 1024 F %L7 P 2048"



0.10 ETAT DES PARTITIONS : EXTRACTEUR.  
0.20 MESURE DU TEMPS D'EXECUTION.

```

1.10 S INT=^MON(0),ST=$V(44),JT=$V(ST),DT=$V(ST+4),J1=JT+18
1.12 S T1=ST+28,T2=ST+30,L=0,CHG=1
1.15 S J2=JT+82,J3=JT+102,T3=ST+26,T4=ST+154,KA=1
1.20 G 3:L>31 S A="",B="" I $V(65400)/100&.64=.64 S ^(.5)=KA-1 Q
1.25 H 0,0,0 S L=L+1 D 5
1.30 S T=$T,G=$V(T3),WJ=$V(T4) F I=0,2,46,62 S DT(I)=$V(DT+I)
1.40 F I=JT:2:J1,J2:2:J3 S JT(I-JT/2)=$V(I)
1.45 S L=L+1 D 5
1.50 F I=0:1:9 S A=A@JT(I)@"*"
1.60 F I=0,2,46,62 S A=A@DT(I)@"*"
1.70 S A=A@T@"*"@G@"*"@WJ@"*" F I=41:1:51 S B=B@JT(I)@"*"
1.80 S ^ (KA)=A,^(KA+1)=B,KA=KA+2,L=L+1,N=INT-($T-T) S:N<0 N=1 D 5 H N G 1.2

3.10 A 3 T #,!!,"CHARGE DU SYSTEME : ",CHG,!!!
3.20 T ?8,"$T",?19,"$28",?31,"$30",?43,"$31",!!
3.30 F L=1:1:30 T ?5,T(L),?20,A(L),?32,B(L),?43,C(L),!
3.40 I CHG<7 S %L9 D 3.6 S CHG=CHG+1,L=0 G 1.25 ← CHARGE DU SYSTEME
3.50 U 3 Q
3.60 F M=1:1:30 K T(M),A(M),B(M),C(M)

5.10 S T(L)=$T,A(L)=$V(T1)/100&2.55*100
5.20 S B(L)=$V(T2)/100&2.55*100,C(L)=$V(T2)/25600*100 ← PRELEVEMENT
DU TEMPS.

200.10 T "%L6" R X D "P 1024.F %L6 P 2048"

```

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



%B3

24/7/1975

12.96

%B3

0.01 CHARGE DU SYSTEME (ALL RESOURCES).

1.10 I \$V(65400)/100&.64= .64 A 1 T "%B3:EXIT",! U 1 H

1.20 F J=1:1:10 D 1.4 H 1

1.30 H 1 G 1.1

1.40 F K=1:1:3 S DCA(K)=^ACD(1,1,K) D 2

2.01 F I=1:1:3 D 2.1,2.2

2.05 Q

2.10 S T=\$T, A=\$V(\$V(44)+28)/100&2.55\*100

2.20 S B=\$V(\$V(44)+30)/100&2.55\*100, C=\$V(\$V(44)+30)/25600\*100

200.10 T "%B3" R X D "P 1024 F %B3 P 2048"

INTERACTIVE SYSTEMS N.V.

BRUSSELS, BELGIUM



%L8

24/7/1975 12.96

%L8

0.01 CHARGE DU SYSTEME (CPU ONLY).

1.10 I \$V(65400)/100&.64=.64 H  
1.20 S ST=\$V(44), T28=ST+28, T30=ST+30  
1.30 F I=1:1:60 D 1.5, 1.6  
1.40 G 1.1  
1.50 S T(I)=\$T, A(I)=\$V(T28)/100&2.55\*100  
1.60 S B(I)=\$V(T30)/100&2.55\*100, C(I)=\$V(T30)/25600\*100  
  
2.10 A 3 T #  
2.20 F J=1:1:I T ?5, T(I), ?15, A(I), ?25, B(I), ?35, C(I), !  
  
200.10 T "%L8" R X D "P 1024 F %L8 P 2048"

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



%L9

24/7/1975 12.96

%L9

0.01 CHARGE DU SYSTEME (DISK ACCESS ONLY).

1.05 A 1 T "PROGRAM START", ! U 1  
1.10 I \$V(65400)/100&.64=.64 A 1 T "CHARGE EXIT", ! U 1 H  
1.20 F J=1:1:10 S DCA(J)=^ACD(1,1,J)  
1.30 F K=1:1:10 S ^ACD(1,1,K)=DCA(K)  
1.40 G 1.1

200.10 T "%L9" R X D "P 1024 F %L9 P 2048"

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



%L5 1/8/1975 10.74

%L5

0.01 OBSERVATION DU TEMPS.

1.10 S ST=\$V(44), T28=ST+28, T30=ST+30 I=1  
1.20 S A(I)=\$V(T28)/100&2.55\*100, B(I)=\$V(T30)/100&2.55\*100  
1.30 S C(I)=\$V(T30)/25600\*100, T(I)=\$T, I=I+1 Q: I>120 G 1.2  
200.10 T "%L5" R X D "P 1024 F %L5 P 2048"

INTERACTIVE SYSTEMS N.V.  
BRUSSELS, BELGIUM



%L4 1/8/1975 10. 75

%L4

0. 10 OBSERVATION DU TEMPS.

1. 10 A 3 T ?8, "\$T", ?19, "S28", ?31, "S30", ?43, "S31", !!!  
1. 20 S ST=\$V(44), T28=ST+28, T30=ST+30  
1. 30 S A=\$V(T28)/100&2. 55\*100, B=\$V(T30)/100&2. 55\*100  
1. 40 S C=\$V(T30)/25600\*100, T=\$T  
1. 50 T ?5, T, ?20, A, ?32, B, ?43, C, ! G 1. 3

200. 10 T "%L4" R X D "P 1024 F %L4 P 2048"

INTERACTIVE SYSTEMS N.V.  
BRUSSELS, BELGIUM



%B1 1/8/1975 10.75

%B1

0.10 OBSERVATION DU TEMPS.

1.10 S ST=\$V(44), T28=ST+28, T30=ST+30, I=0, ^T(0)=I  
1.20 S ^(I+1)=\$T, ^(I+2)=\$V(T28)/100&2.55\*100  
1.30 S ^(I+3)=\$V(T30)/100&2.55\*100, ^(I+4)=\$V(T30)/25600\*100  
1.40 G 1.5: I>200 S I=I+4 G 1.2  
1.50 S ^ (0)=200 @

200.10 T "%B1" R X D "P 1024 F %B1 P 2048"

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



## 0.10 MESURE DU TEMPS DE REPONSE.

1.05 S ST=\$V(44), TS=\$V(ST+32)/100&2.55\*100, S28=ST+28, CHG=1, ACC=1, B="" A 3  
 1.10 T #, !!!!!!!!, "VALEUR D'UNE TRANCHE DE TEMPS STANDARD : ", TS, " TICKS.  
 1.20 T #, !, "TYPE D'ACCES : ", ACC, ! T !!!!!, "PROCEDURE D'ACCES. ", !!!  
 1.30 S H=9.1+ACC T \$S(H), ! T #, !!  
 1.40 G 2

2.10 T "CHARGE DU SYSTEME : ", CHG, " PARTITION", B, !! F I=1:1:53 D 3  
 2.20 S SUM=0 F I=3:1:53 S SUM=SUM+DIF(I)  
 2.30 S MOY=SUM/50, ^TAB(ACC, CHG)=MOY T "TEMPS MOYEN : ", MOY, !  
 2.40 F M=0:1:4 T ! F N=3:1:12 T DIF(M\*10+N), "/"  
 2.50 T !!! S CHG=CHG+1 I CHG=2 S %B3(6) S B="S" G 2.1  
 2.60 I CHG>8 G 2.8  
 2.70 S %B3 G 2.1 ← CHARGE DU SYSTEME.  
 2.80 D 6 A 3 S CHG=1, B="", ACC=ACC+1 I ACC<=4 D 1.2, 1.3 G 2  
 2.90 G 4

## → CALCUL DU TEMPS DE REPONSE.

3.10 S T(2)=\$V(S28)/100&2.55\*100, T(1)=\$T D 9+ACC  
 3.20 S T(4)=\$V(S28)/100&2.55\*100, T(3)=\$T, Y=(T(1)\*100)+((50-T(2))\*2)  
 3.30 S X=(T(3)\*100)+((50-T(4))\*2), DIF(I)=X-Y I DIF(I)<0 S DIF(I)=DIF(I)+100  
 3.40 Q

4.10 T #, !!!!!, "EN ABSCISSE: CHARGE DU SYSTEME. ", !  
 4.20 T "EN ORDONNEE: TYPE D'ACCES DISQUE. ", !!!!! S STR=".....\*"  
 4.30 T !, ?5, "\*" F I=1:1:8 T STR  
 4.40 T !, ?5, "\*" F I=1:1:8 T " ", I, " \*"  
 4.50 S STR="\*\*\*\*\*" D 4.3 F L=1:1:ACC-1 S X=^TAB(L, 1) D 5  
 4.60 T # U 3 Q

5.10 T !, ?2, L, ?5, "\*" F K=1:1:8 D 5.3  
 5.20 S STR=".....\*" D 4.3 Q  
 5.30 S Y=^(K)@, INT=6+((K-1)\*8)+(6-\$L(Y)) T ?INT, Y, " \*"

6.01 A 0 F I=1:1:1000 P 7  
 6.05 T ! "PLEASE SET BIT 6 OF SWITCH REGISTER."  
 6.10 T ! "THE MESSAGE 'XB3:EXIT' MUST APPEAR 7 TIMES ON THE CONSOLE"  
 6.20 T ! "AFTER THAT, PLEASE RESET BIT 6 OF SWITCH REGISTER"  
 6.25 R ! "TYPE A 'CR'", S6  
 6.30 T ! "OK, IT'S OVER, THANK YOU" U 0

10.10 S Z=^REP(1)

11.10 S Z=^REP(1, 1)

12.10 S Z=^REP(1, 1, 1)

13.10 S Z=^REP(1, 1, 1, 1)

200.10 T "%B4" R X D "P 1024 F %B4 P 2048"



%M8 1/8/1975 10. 77

%M8

0. 10 CHARGE DU SYSTEME (ALL RESOURCES)  
0. 20 CHARGE DE TYPE B.

1. 10 I \$V(65400)/100&. 64=. 64 A 1 T "%M8:EXIT", ! U 1 H  
1. 20 F J=1:1:10 S DCA(J)=^ACD(1,1,J) H 1 D 2  
1. 30 H 5 G 1. 1

2. 01 F K=1:1:5 D 2. 1, 2. 2  
2. 05 Q

2. 10 S T=\$T, A=\$V(\$V(44)+28)/100&2. 55\*100  
2. 20 S B=\$V(\$V(44)+30)/100&2. 55\*100, C=\$V(\$V(44)+30)/25600\*100

200. 10 T "%M8" R X D "P 1024 F %M8 P 2048"

INTERACTIVE SYSTEMS N.V./  
BRUSSELS, BELGIUM



%M9 1/8/1975 10. 76

%M9

0. 10 CHARGE DU SYSTEME (ALL RESOURCES)  
0. 20 CHARGE DE TYPE C.

1. 10 I \$V(65400)/100&. 64=. 64 A 1 T "%M9:EXIT", ! U 1 H  
1. 20 F J=1:1:10 S DCA(J)=^ACD(1,1,J) H 2 D 2  
1. 30 H 5 G 1. 1

2. 01 F K=1:1:5 D 2. 1. 2. 2  
2. 05 Q

2. 10 S T=\$T, A=\$V(\$V(44)+28)/100&2. 55\*100  
2. 20 S B=\$V(\$V(44)+30)/100&2. 55\*100, C=\$V(\$V(44)+30)/25600\*100

200. 10 T "%M9" R X D "P 1024 F %M9 P 2048"

INTERACTIVE SYSTEMS N.V.  
BRUSSELS, BELGIUM